MATLAB 命令大全

姚 东 王爱民
冯 峰 王朝阳
等 编著

人民邮电出版社

内容提要

MATLAB 是当前最流行的大型数学工具软件之一。本书全面介绍了最新的 MATLAB 5.3 版的全部命令的描述、语法和应用。包括：数值计算、数据分析、绘图和图形图像处理、用户图形界面创建等几大类。各个命令都附有实例，易学易用。

本书可供在机械、电子、通信、医疗、建筑等领域处理数值计算和分析的用户查阅、参考。

MATLAB 命令大全

◆ 编　著　姚　东　王爱民　冯　峰　王朝阳　等

责任编辑　张立科

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号

北京汉魂图文设计有限公司制作

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092　1/16

印张：38.25

字数：963 千字　　2000 年 6 月第 1 版

印数：1－5 000 册　　2000 年 6 月北京第 1 次印刷

ISBN 7-115-08558-7/TP·1654

定价：56.00 元

前　言

MATLAB 是当前最流行的、功能强大的科技应用软件和编程语言之一，它集数学计算、可视化和可编程等功能于一体，而且易学易用，具有非常广泛的应用前景。

MATLAB 从推出到今天的 5.3 版，功能日益增强，逐渐发展成为一个大家族，包含众多的功能各异的工具箱。这些工具箱是 MATLAB 函数的在各个具体领域扩展的集合，这些领域涉及信号处理、控制系统、神经网络、模糊逻辑和仿真等。

整个 MATLAB 系统有五个主要的部分：(1)MATLAB 语言。它是基于矩阵/数组的高级语言，包括流程控制语句、函数、数据结构和输入/输出等，它还具有面向对象编程的特点。它既适合编写小巧玲珑的程序，也适用于开发复杂的大型应用程序。(2)MATLAB 工作环境。它集成了一系列的工具和应用，方便用户管理环境变量，输入/输出数据，开发、管理、调试用户自己的 M 文件以及 MATLAB 的应用程序。(3)图形处理。它既包括高层的二维和三维的数据可视化、图像处理、动画的指令，也包括低层的绘图指令，允许用户为应用程序设计自己的用户图形界面。(4)MATLAB 数学函数库。它包括数量庞大的计算函数，从简单的基本函数，如求和、正弦和复数运算到复杂的矩阵求逆，矩阵的特征值，贝塞尔函数和快速傅立叶变换等。(5)MATLAB 应用程序界面（API）。它是一组动态的库函数，使得用户在自己的 C 和 Fortran 程序中可以和 MATLAB 交互，调用 MATLAB 的动态连接库作计算。

本书对 MATLAB 的每个函数或指令的含义进行了描述，对其语法格式和参数作了说明，并尽可能对每个指令都给出实例供参考。为方便阅读和检索，所有的函数或指令分为以下几个主要方面介绍：通用指令，运算符和特殊算符，逻辑运算函数，语言结构和调试指令，基本矩阵和操作，特殊矩阵，基本数学函数，特殊数学函数，坐标系变换函数，矩阵函数和数值线性代数，数据分析和 Fourier 变换，多项式和插值函数，稀疏矩阵，声音处理函数，字符串函数，低层文件输入输出函数，Bitwise Functions，结构函数，对象函数，Cell Array Functions，多维数组函数，绘图和数据可视化函数，图形用户界面创建函数等。

本书的主要读者对象是：科研人员，工程技术人员，理工科高等院校的教师、研究生和本科生。

本书主要由姚东、王爱民、王朝阳、冯峰编写。另外，参加本书编写和整理的人员还有：马国栋、周建、王美芬、陆天明、严春、马大洪、杜少军、肖文、柳冰、常锦、沈丽丽、迟瑞、郭讯、陈显、唐永、于文兵、秦芳芳、康顺义、金大坚、苏林、高廉、蒋进武、韩金龙、薛升、罗成贵、齐振宇、李成等。

由于 MATLAB 功能强大，内容众多，涉及领域广泛，加之编者水平有限，书中错误之处在所难免，希望广大读者不吝赐教。

作者

2000 年 2 月

目 录

第一章 常用命令

1.1 管理命令和函数

1. 添加目录

名称：addpath

添加目录到 MATLAB 的搜索路径。

语法：该函数有如下几种表达形式：

- addpath('directory')
- addpath('dir1','dir2','dir3',...)
- addpath(...,'-flag')

描述：addpath('directory')命令将添加指定目录到 MATLAB 当前的搜索路径中。

addpath('dir1','dir2','dir3',...)命令将添加所有的指定目录到 MATLAB 当前的搜索路径中。

addpath(...,'-flag')命令将按照 flag 的值挂起或添加所有的指定目录到 MATLAB 当前的搜索路径中。参数 flag 的值见表 1-1。

表 1-1 参数 flag 的值

flag 的值	说明
0(begin)	挂起指定的目录
1(end)	添加指定的目录

2. 显示 HTML 文档

名称：doc

在 Web 浏览器上显示 HTML 文档。

语法：该函数有如下几种表达形式：

- doc
- doc function
- doc toolbox/function

描述：doc：显示 MATLAB 帮助桌面。

doc function：显示指定 MATLAB 功能函数的 HTML 帮助文档。

doc toolbox/function：显示指定工具箱中函数的 HTML 帮助文档。

3. 帮助文件目录

名称：docopt

定位 UNIX 平台中的帮助文件目录。

语法：该函数有如下两种表达形式：

- docopt
- [doccmd,options,docpath]=docopt

描述：docopt 命令用于显示 UNIX 平台中帮助文件目录的位置。该命令仅用于 UNIX 系统。当用户安装 MATLAB 时，使用该命令指定在线帮助信息的位置，用户可以将在线帮助信息指定在本地的硬盘或光盘上。当用户需要重新指定在线帮助信息的位置时，可以在 MATLAB 命令窗口中编辑 docopt.m 文件。

[doccmd,options,docpath]=docopt 用于返回三个字符串：

“doccmd”是指一个字符串，该字符串包含指定显示 HTML 帮助文档的浏览器的命令，其默认值为：

Unix: netscape

Windows: -na-

Macintosh: -na-

“options”是一个包含附加的配置选项的字符串，其默认值为：

Unix: ''

Windows: -na-

Macintosh: -na-

“docpath”是一个包含指向 MATLAB 在线文档的路径的字符串。如果该字符串为空值，则 doc 命令将自动计算到 MATLAB 在线文档的路径。若要全部替换在线帮助文件目录的位置，需更新$MATLAB/toolbox/local/docopt.m。

若要覆盖全部设置，需将

$MATLAB/toolbox/local/docopt.m

复制到

$HOME/matlab/docopt.m

并在此处进行改动。为了使改动生效，$HOME/matlab 必须在用户的 MATLAB 路径中。

4. *帮助*

名称：help

显示 MATLAB 命令和 M 文件的在线帮助。

语法：该函数有如下两种表达形式：

- help
- help topic

描述：help 将列出所有主要的帮助主题。每个帮助主题与 MATLAB 搜索路径中的一个目录名相对应。

help topic 将给出指定主题的帮助。主题可以是一个函数名、目录名或 MATLAB 的局部路径名。如果指定的主题是一个函数名，将显示该函数的相关信息；如果指定的主题是一个目录名，将显示指定目录的内容文件。使用该命令时，用户不必给出指定目录的全路径名，只需要给出目录的最后一个部分或后边的几个部分，就可以查询到相关的主题。

MATLAB 帮助系统和 MATLAB 本身一样，都是高度可扩展的。通过与 MATLAB 中 M 文件和工具箱使用自有文档相同的方法，用户可以为自己的 M 文件和工具箱书写帮助文档。通过显示 MATLAB 搜索路径的每个目录中内容文件的第一行(H1 行)，help 命令列出了全部

的帮助主题。内容文件是指每个目录中名为“Contents.m”的 M 文件。

对于命令 help topic，当 topic 是一个目录名时，显示该目录 M 文件中的内容。若一个内容文件不存在，该命令将显示该目录中所有文件的 H1 行；当 topic 是一个函数名时，则列出“topic.m”文件中最接近的注释行。

举例：

在 MATLAB 命令窗口中输入：

help general

将显示 matlab\general 目录中的 M 文件的内容。

在 MATLAB 命令窗口中输入：

help sin

将显示该 M 文件中最接近的注释行。

```
SIN      Sine.
    SIN(X) is the sine of the elements of X.
Overloaded methods
    help sym/sin.m
```

5. 帮助桌面

名称： helpdesk

在浏览器中显示帮助桌面网页，为用户提供更加详尽的帮助。

语法： helpdesk

描述： 在 MATLAB 命令窗口中执行 helpdesk 命令将通过浏览器打开帮助桌面网页，通过帮助桌面网页，用户可直接访问在线帮助的全部库，其中包括 MATLAB 参考及用户手册。

在 Windows 平台上，用户还可以直接在 MATLAB 界面的“Help”菜单中选择“Help Desk”来访问帮助桌面。

举例：

在 MATLAB 命令窗口中输入：helpdesk

结果如图 1-1 所示。

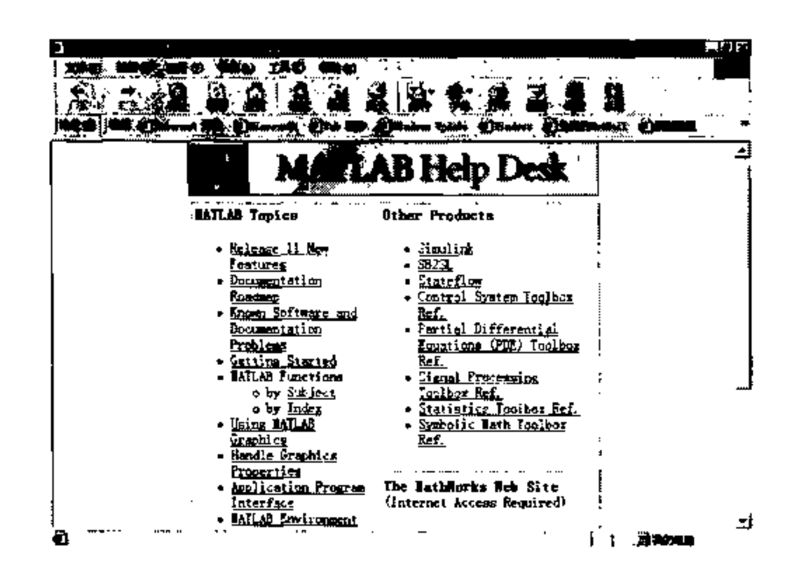


图 1-1 MATLAB 帮助桌面

6. 显示帮助窗口

名称：helpwin

显示帮助窗口，该帮助窗口提供所有命令的帮助。

语法：该函数有如下两种表达形式：

- helpwin
- helpwin topic

描述：helpwin 显示帮助窗口，在该窗口中列出了所有按照主题分组的命令。通过帮助窗口，用户既可以查看命令的摘要描述，也可以获得进一步的帮助信息。

在 Windows 平台上，用户还可以直接在 MATLAB 界面的“Help”菜单中选择“Help Window”选项来打开帮助窗口，单击工具条上的“问号”按钮也可以打开帮助窗口。

举例：

在 MATLAB 命令窗口中输入 helpwin winfun，将显示 winfun 的帮助窗口，如图 1-2 所示。

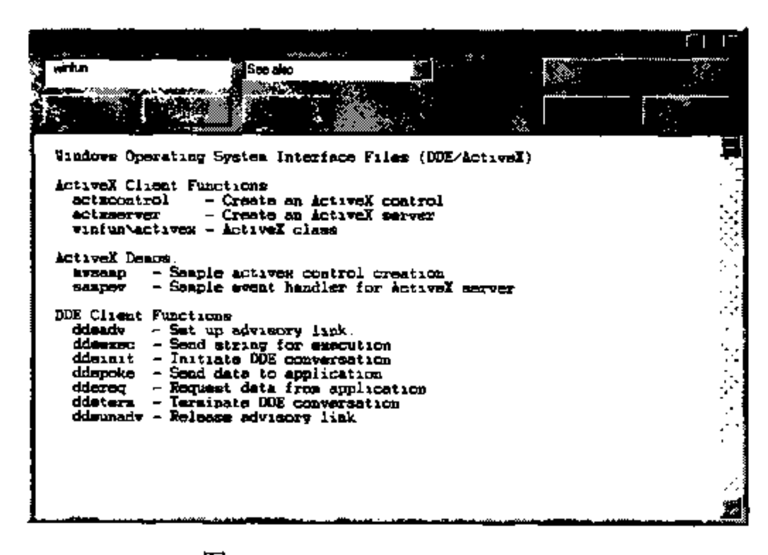


图 1-2 MATLAB Help Windows

7. 异常信息

名称：lasterr

最后异常信息。

语法：该函数有如下两种表达形式：

- str = lasterr
- lasterr('')

描述：语句 str = lasterr 可以返回 MATLAB 中产生的最后一个异常信息。

语句 lasterr('')将重新设置 lasterr 并返回一个空的矩阵，直到下一个异常发生。

举例：

下面一段代码实现了一个能够检测到 MATLAB 中的异常的函数：

```
function catchfcn
m = lasterr;
n = findstr(m,'Inner matrix dimensions');
```

```
if n~=[]
    disp('Wrong dimensions for matrix multiply')
else
    k = findstr(m,'Undefined function or variable')
    if (k~=[])
        disp('At least one operand does not exist')
    end
end
```

该函数能够处理两种矩阵相乘产生的异常情况。

8. 警告信息

名称：lastwarn

最后的警告信息。

语法：该函数有如下几种表达形式：

- lastwarn
- lastwarn('')
- lastwarn('string')

描述：命令 lastwarn 返回一个包含 MATLAB 产生的最后一个警告信息的字符串。

lastwarn('')将重置 lastwarn 函数，返回一个空的字符串矩阵，直到下一个警告产生为止。

lastwarn('string')将最后一个警告信息设置为括号中的字符串。

9. 搜索关键字

名称：lookfor

在所有的帮助条目中搜索关键字。

语法：该函数有如下两种表达形式：

- lookfor topic
- lookfor topic -all

描述：lookfor topic 在基于 MATLAB 搜索路径的所有 M 文件中搜索 H1 行，该命令将把在搜索中发现与关键字“topic”相匹配的所有 H1 行都显示在 MATLAB 的命令窗口中。

lookfor topic –all 将根据关键字“topic”在所有的 M 文件中进行搜索。

举例：

分别在 MATLAB 命令窗口中输入：

```
lookfor simulink
lookfor simulink -all
```

通过在命令窗口中的输出结果，用户可以更好地理解这两条命令的区别。

10. 部分路径名

名称：partialpath

部分路径名。

描述：部分路径名是一个 MATLAB 相对路径名，常用于查找私有文件和方法文件，而私有文件和方法文件通常是隐藏的；部分路径名还用于当使用所给文件名搜索到不止一个文件时来限制那些搜索到的文件，可以使用户更容易地查找到 MATLAB 的工具箱及相关文件。

11. 目录搜索路径

名称：path

管理 MATLAB 的目录搜索路径。

语法：该函数有如下几种表达形式：

- path
- p = path
- path('newpath')
- path(path,'newpath')
- path('newpath',path)

描述：path 将在 MATLAB 的命令窗口中打印出 MATLAB 当前搜索路径的设置状况。

p = path 将把 MATLAB 当前搜索路径保存到一个字符串变量“p”中。

path('newpath')将当前路径改变为新路径名。

path(path,'newpath')将添加一个新的目录到当前路径中。

path('newpath',path)将准备把一个新的路径添加到当前路径中。

举例：

下面是在不同的操作系统中添加一个新的目录到搜索路径中的例子：

```
UNIX      path(path,'/book/matlab')
VMS       path(path,'DISKS1:[BOOK.MATLAB]')
Windows   path(path,'BOOK\MATLAB')
```

12. 路径浏览器

名称：pathtool

启动路径浏览器，一个查看和更改 MATLAB 路径的图形用户界面。

语法：pathtool

描述：使用 pathtool 命令将打开路径浏览器，用户可以通过路径浏览器查看或改变 MATLAB 的搜索路径。在路径浏览器中用户可以：

(1) 在指定路径前添加一个目录。

(2) 从路径中删除一个目录。

(3) 将设置保存到“pathdef.m”中。

(4) 恢复默认设置。

在 Windows 操作系统中，用户可以用几种不同的方法来启动路径浏览器：

(1) 在“File”菜单中选择“Set Path”。

(2) 直接在工具栏中单击 Path Browser 按钮。

(3) 在“Editor/Debugger”中，从“View”菜单中选择“Path Browser”。

13. MATLAB 编译器

名称：profile

启动 MATLAB 编译器，这是一个用于调试和优化 MATLAB 代码的工具。

语法：该函数有如下几种表达形式：

- profile on
- profile on -detail level

- profile on -history
- profile off
- profile resume
- profile clear
- profile report
- profile report basename
- profile plot
- profile status
- stats = profile('info')

描述：MATLAB profiler 通过跟踪执行时的 M 文件代码来帮助用户进行调试和优化工作。对于 MATLAB 的每一个函数，profiler 将记录其执行时间、调用次数、父函数及子函数。

profile on 命令将启动 MATLAB profiler，并清除以前的记录。

profile on -detail level 命令将启动 MATLAB profiler，清除以前的记录，并对用 level 指定的函数集产生记录。

参数 level 的值见表 1-2。

表 1-2　　参数 level 的值

Level 的值	说明
mmex	M 函数，M 子函数，MEX 函数
builtin	内置函数
operator	操作符函数

profile on –history 命令将启动 MATLAB profiler，清除以前的记录。

profile off 命令中止 MATLAB profiler。

profile resume 命令将重新启动 MATLAB profiler，并且不清除以前的记录统计表。

profile clear 命令清除 MATLAB profiler 的统计记录。

profile report 命令将中止 MATLAB profiler，产生一个 HTML 格式的概貌报告，并在用户的 Web 浏览器中显示该报告。

profile report basename 命令将中止 MATLAB profiler，产生一个 HTML 格式的概貌报告，把该报告使用指定文件名保存在当前目录下，并在用户的 Web 浏览器中显示该报告。

profile plot 命令将终止 MATLAB profiler，并显示一个图形窗口。

profile status 命令将显示一个包含当前 profiler 状态的结构。该结构的字段见表 1-3。

表 1-3　　profiler 状态结构的字段

字段	值
ProfilerStatus	'on'或'off'
DetailLevel	'mmex'，'builtin'或'operator'
HistoryTracking	'on'或'off'

stats = profile('info')命令将终止 profiler，并显示一个包含 profiler 中结果的结构。用户可以使用该命令来访问由 profiler 产生的数据。

举例：

在 MATLAB 命令窗口中输入：

profile on

profile report

结果如图 1-3 所示。

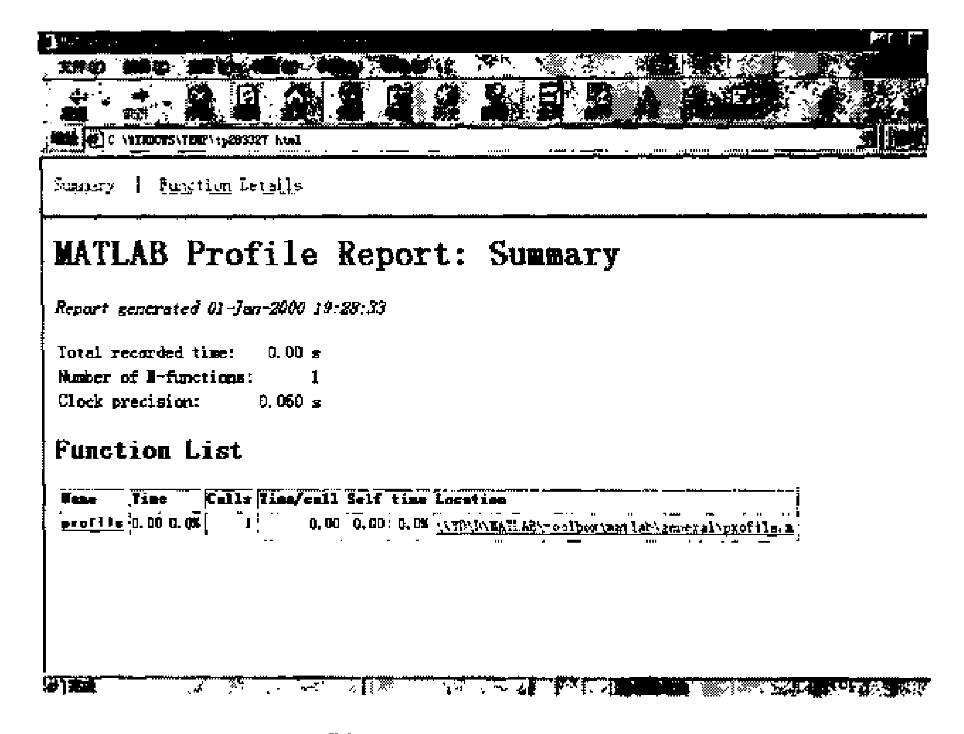


图 1-3 MATLAB 概貌报告

继续输入：

profile status

结果为：

ans =

ProfilerStatus: 'off'

DetailLevel: 'mmex'

HistoryTracking: 'off'

14. PROFILE 报告

名称：profreport

产生一个 PROFILE 报告。

语法：该函数有如下几种表达形式：

- profreport
- profreport(basename)
- profreport(stats)
- profreport(basename,stats)

描述：profreport 命令将暂停执行 profiler，使用 profiler 当前结果产生一个用 HTML 文档格式描述的概貌报告，并且在用户的 Web 浏览器上显示该报告。

profreport(basename)命令将暂停执行 profiler，使用 profiler 当前结果产生一个用 HTML 文档格式描述的概貌报告，并且使用用户提供的基本文件名在用户的 Web 浏览器上显示该报告。由于产生的报告包含不止一个文件，所以不要为基本文件名提供扩展名。

profreport(stats)命令将暂停执行 profiler，使用 profiler 当前结果产生一个用 HTML 文档格式描述的概貌报告，并且在用户的 Web 浏览器上显示该报告。stats 是指使用 stats=profile('info')返回的编译器信息结构。

profreport(basename,stats)命令将暂停执行 profiler，使用 profiler 当前结果产生一个用 HTML 文档格式描述的概貌报告，并且使用用户提供的基本文件名在用户的 Web 浏览器上显示该报告。stats 是指使用 stats=profile('info')返回的编译器信息结构。由于产生的报告包含不止一个文件，因此不要为基本文件名指定扩展名。

举例：

(1) 为计算 Lotka-Volterra 人口模型的代码运行编译器。

在 MATLAB 命令窗口中输入如下命令：

```
profile on -detail builtin –history
[t,y] = ode23('lotka',[0 8],[10;10]);
```

(2) 查看包含编译器中结果的结构。

在 MATLAB 命令窗口中输入如下命令：

```
stats = profile('info')
```

MATLAB 的命令窗口中将返回如下结果：

```
stats =
        FunctionTable: [28x1 struct]
        FunctionHistory: [2x2944 double]
        ClockPrecision: 0.0500
```

(3) 查看函数表结构的第三个元素的内容。

在 MATLAB 命令窗口中输入如下命令：

```
stats.FunctionTable(3)
```

MATLAB 的命令窗口中将返回如下结果：

```
ans =
             FunctionName: 'ode23'
                MfileName: 'D:\MATLAB\toolbox\matlab\funfun\ode23.m'
                     Type: 'M-function'
                 NumCalls: 1
                TotalTime: 0.7700
       TotalRecursiveTime: 0.7700
                 Children: [21x1 struct]
                  Parents: [0x1 struct]
            ExecutedLines: [167x3 double]
```

(1) 根据结构显示概貌报告。

在 MATLAB 命令窗口中输入如下命令：

```
profreport(stats)
```

MATLAB 将在用户的 Web 浏览器上显示该概貌报告，如图 1-4 所示。

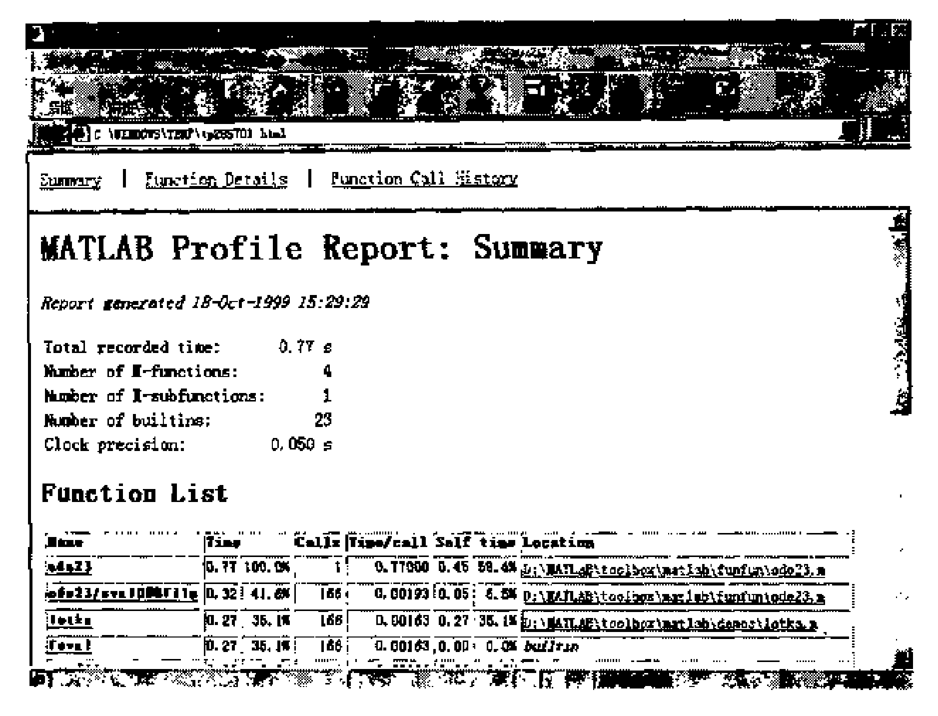


图 1-4 根据 profiler 中的结果显示的 HTML 格式报告

15. 删除指定目录

名称：rmpath

从 MATLAB 搜索路径中删除指定目录。

语法：rmpath directory

描述：rmpath directory 命令将从 MATLAB 的当前搜索路径中删除指定目录。此外，MATLAB 也可以接受如下的命令格式：rmpath('directory')

举例：

可以使用如下命令删除目录/local/matlab/myToolbox：

rmpath /local/matlab/myToolbox

16. 显示指定文件的内容

名称：type

显示指定文件的内容。

语法：type filename

描述：type filename 将根据所给的全路径名或相对局部路径名在 MATLAB 命令窗口中显示指定文件的内容。

如果用户没有指定扩展名，缺省情况下“type”命令将自动为文件添加“.m”的扩展名。“type”命令将在 MATLAB 搜索路径中检查指定目录，这样便于在计算机显示屏幕上列出指定的 M 文件的内容。

举例：

在 MATLAB 命令窗口中输入 type tool1.bar

将显示 tool1.bar 的内容。

在 MATLAB 命令窗口中输入 type tool1

将显示 tool1.m 的内容。

17. 版本信息

名称：ver

显示 MATLAB、SIMULINK 及工具箱的版本信息。

语法：该函数有如下两种表达形式：

- ver
- ver toolbox

描述：ver 命令将显示 MATLAB、SIMULINK 及工具箱的版本信息。

ver toolbox 将显示指定工具箱的版本号和发布日期。

用户可以通过查看 ver.m 文件来获得如何使用 ver 命令的相关信息。

举例：

在 MATLAB 命令窗口中输入命令 ver

将显示用户的显示 MATLAB、SIMULINK 及工具箱的版本信息：

```
-------------------------------------------------
MATLAB Version 5.3.0.10183 (R11) on PCWIN
MATLAB License Number: 134511
-------------------------------------------------
MATLAB Toolbox                              Version 5.3    (R11)   15-Jan-1999
Symbolic Math Toolbox                       Version 2.1    (R11)   11-Sep-1998
NAG Foundation Toolbox - Numerical ...      Version 1.0.3  (R11)   06-Jun-1998
Partial Differential Equation Toolbox       Version 1.0.3  (R11)   21-Nov-1997
Statistics Toolbox                          Version 2.2    (R11)   24-Jul-1998
Signal Processing Toolbox                   Version 4.2    (R11)   10-Jul-1998
Control System Toolbox                      Version 4.2    (R11)   15-Jul-1998
SystemBuild to Simulink Translator          Version 2.0    (R11)   10-Aug-1998
Stateflow                                   Version 2.0    (R11)   16-Jan-1999
Simulink                                    Version 3.0    (R11)   01-Sep-1998
MATLAB Tour                                 Version 1.2    (R11)   04-Sep-1998
```

1. MATLAB 版本号

名称：version

该命令将返回用户的 MATLAB 版本号。

语法：该函数有如下两种表达形式：

- v = version
- [v,d] = version

描述：v = version 命令将把 MATLAB 的版本号保存在字符串变量“v”中。

[v,d] = version 命令除了把 MATLAB 的版本号保存在字符串变量“v”中外，还将把 MATLAB 的发布日期保存到字符串变量“d”中。

举例：

在 MATLAB 命令窗口中输入 v = version

将显示：

v =

5.3.0.10183 (R11)

在 MATLAB 命令窗口中输入[v,d] = version

将显示：

v =

5.3.0.10183 (R11)

d =

Jan 21 1999

19. 打开指定的文件或 Web 站点

名称：web

使用 Web 浏览器打开指定的文件或 Web 站点。

语法：该函数有如下两种表达形式：

web url

stat = web(...)

描述：web url 命令将根据所给 URL 来打开指定的文件或 Web 站点。

stat = web(...)把“web”命令的执行状态保存到变量“stat”中，变量“stat”的值见表 1-4。

表 1-4　　变量“stat”的值

值	作用
0	成功找到浏览器
1	没有找到浏览器
2	找到浏览器但无法启动

举例：

若需要打开 C:\My Documents\MATLAB\matlab1\text.htm 文件，则可以在 MATLAB 命令窗口中直接输入：web C:\My Documents\MATLAB\matlab1\text.htm

若需要打开某个 Web 站点(例如 MathWorks 公司的站点)，则可以在 MATLAB 命令窗口中直接输入：web http://www.mathworks.com

若需要直接给某个 Web 站点发送电子邮件，则可以在 MATLAB 命令窗口中直接输入：

web mailto:email_address

20. M 文件、MAT 文件和 MEX 文件

名称：what

列出当前目录中所有的 M 文件、MAT 文件和 MEX 文件。

语法：该函数有如下几种表达形式：

- what
- what dirname
- what('dirname')

描述：what 命令将列出当前目录中所有的 M 文件、MAT 文件和 MEX 文件。

what dirname 命令将根据所给的目录名列出该目录中所有的 M 文件、MAT 文件和 MEX 文件。通常不必给出全路径名。

what('dirname')命令将返回一个由数个字段组成的结构数组。结构数组的字段见表 1-5。

表 1-5　　结构数组的字段

字段	描述
Path	目录的路径
M	M 文件名的单元数组
MAT	MAT 文件名的单元数组
MEX	MEX 文件名的单元数组
MDL	MDL 文件名的单元数组
P	P 文件名的单元数组
Classes	类文件名的单元数组

举例：

在 MATLAB 的命令窗口中输入 what ops 或 what matlab\ops

结果都将列出 MATLAB 的"ops"目录中的所有 M 文件。

在不同的操作系统中使用该命令的格式不同，对于上面的例子，不同的操作系统的输入格式如下：

UNIX　　matlab/ops

VMS　　MATLAB.OPS

Windows　MATLAB\ops

21. MATLAB 及工具箱的"README"文件

名称：whatsnew

显示 MATLAB 及工具箱的"README"文件。

语法：该函数有如下几种表达形式：

- whatsnew
- whatsnew matlab
- whatsnew toolboxpath

描述：whatsnew 显示 MATLAB 产品及指定工具箱的"README"文件的摘要。

whatsnew matlab 显示 MATLAB 的"README"文件。

whatsnew toolboxpath 显示字符串"toolboxpath"指定的工具箱的"README"文件。

举例：

查看信号处理工具箱的"README"文件，可以在 MATLAB 的命令窗口中直接输入：

whatsnew signal

查看控制工具箱的"README"文件，可以在 MATLAB 的命令窗口中直接输入：

whatsnew control

22. 查找函数和文件的位置

名称：which

查找函数和文件的位置。

语法：该函数有如下几种表达形式：

- which fun
- which fun -all
- which file.ext
- which fun1 in fun2
- which fun(a,b,c,...)

- s = which(...)

描述：which fun 命令将显示指定函数的全路径名。被指定的函数可以是一个 M 文件、MEX 文件、工作空间变量、内置函数或 SIMULINK 模型。对于指定的函数为工作空间变量、内置函数或 SIMULINK 模型的情况，将显示一条信息来标明。

which fun –all 命令将显示所有使用指定名称的函数的路径。在给出的列表中最先列出的是使用 which fun 命令显示的结果。“-all”标记能够在 which 命令的所有形式中使用。

which file.ext 命令将显示指定文件的全路径名。

which fun1 in fun2 命令将在名为“fun2”的 M 文件上下文中显示名为“fun1”的函数的路径名。这样，当调试“fun2”M 文件时，将对“fun1”文件做同样的处理。通过使用该命令，用户可以决定是否使用一个局部或私有的函数来替代 MATLAB 的搜索路径中的函数。

which fun(a,b,c,...)命令将使用所输入的参数来显示指定函数的路径。

s = which(...)命令将把执行“which”命令得到的结果保存到变量“s”中，这样将不会在屏幕上显示所得到的结果。

举例：

在 MATLAB 命令窗口中输入 which sin

将显示：sin is a built-in function.

在 MATLAB 命令窗口中输入 which box

将显示“box.m”的全路径名 D:\MATLAB\toolbox\matlab\graph2d\box.m

1.2 管理变量和工作区

1. 删除所有的内容

名称：clear

从内存中删除所有的内容。

语法：该函数有如下几种表达形式：

- clear
- clear name
- clear name1 name2 name3
- clear global name
- clear keyword

描述：clear 命令将从工作空间内删除所有的变量。

clear name 命令将从工作空间内删除使用“name”指定的变量、M 文件及 MEX 文件。

clear name1 name2 name3 命令将删除工作空间内的“name1”、“name2”和“name3”。

clear global name 从工作空间删除使用“name”指定的全局变量、M 文件及 MEX 文件。

clear keyword 命令将从工作空间内删除指定的关键字。使用“clear keyword”命令删除的关键字的值见表 1-6。

用户可以在“clear”命令中使用通配符“*”。例如，当用户需要删除工作空间内所有以“h”开头的变量，则可以在 MATLAB 命令窗口中输入 clear h*。此外，在 UNIX 系统中，

使用“clear”命令不会影响操作系统分配给 MATLAB 进程的内存数量。

表 1-6 关键字的值

值	说明
functions	从内存中清除当前编译的所有 M 函数
variables	从工作空间内清除所有的变量
mex	从内存中清除所有的 MEX 文件
global	从工作空间内清除所有的全局变量
all	从内存中清除所有的变量、函数及 M 文件，让工作空间为空
classes	从内存中清除所有的变量、函数及 M 文件，让工作空间为空并清除所有类定义

2. 显示文本或数组

名称：disp

显示文本或数组。

语法：disp(X)

描述：disp(X)命令将显示一个不带名称的数组。如果该数组还包含有文本字符串，该字符串将在屏幕上显示。

另一种显示数组的方法是在屏幕上直接输入该数组的名字，例如 X，但这样显示的数组的开头将带有 X =。

举例：

若要显示一个 6 行 4 列的随机数组，可以在 MATLAB 的命令窗口中输入 disp(rand(6,4))

结果为：

```
0.4057    0.3529    0.6038    0.9318
0.9355    0.8132    0.2722    0.4660
0.9169    0.0099    0.1988    0.4186
0.4103    0.1389    0.0153    0.8462
0.8936    0.2028    0.7468    0.5252
0.0579    0.1987    0.4451    0.2026
```

3. 数组的长度

名称：length

数组的长度。

语法：n = length(X)

描述：n = length(X)返回数组 X 的最长维数，若 X 是一个向量，则返回 X 的长度。

举例：

在 MATLAB 命令窗口中输入：

```
x = rand(3,9,6);
n = length(x)
```

结果将显示该数组中最长的维数：

```
n =
    9
```

在 MATLAB 命令窗口中输入一个向量 X：

```
x =
```

```
[1
 2
 4
 8]
n = length(x)
```

结果将显示该向量的长度：

```
n =
    4
```

4. 重新载入变量

名称：load

从磁盘上重新载入变量。

语法：该函数有如下几种表达形式：

- load
- load filename
- load ('filename')
- load filename.ext
- load filename -ascii
- load filename –mat
- S = load(...)

描述：load 从磁盘上重新载入 MATLAB 变量。该命令将载入磁盘上所有以“matlab.mat”格式保存的文件。

load filename 根据给出的全路径名或局部路径名从磁盘上载入所有 MATLAB 变量。

load ('filename')从磁盘上载入名为“filename”的文件。

load filename.ext 从磁盘上读取 ASCII 文件，作为结果的数据将被存入与文件同名的变量中(不带扩展名)。ASCII 文件可以包含 MATLAB 注释(以“%”开头的行)

load filename –ascii 可以将指定名称的文件强行使用 ASCII 文件格式载入。

load filename -mat 可以将指定名称的文件强行使用 MAT 文件格式载入。

S = load(...)把 MAT 文件的内容作为一个结构，而并不直接将该文件载入到 MATLAB 的工作空间内。

MAT 文件是由“save”命令创建并能够被“load”命令读取的二进制双精度 MATLAB 格式的文件。这样就可以在一台计算机上创建并在另外的计算机上读取 MAT 文件。MAT 文件也能够被 MATLAB 以外的其他外部程序操作。

5. 防止 M 文件被删除

名称：mlock

防止 M 文件被删除。

语法：该函数有如下两种表达形式：

- mlock
- mlock(fun)

描述：mlock 命令将对当前正在运行的 M 文件进行锁定，这样可以防止使用“clear”

命令删除 M 文件。

mlock(fun)命令将对内存中指定的 M 文件进行锁定。

6. 允许被锁定的 M 文件被清除

名称：munlock

允许被锁定的 M 文件被清除。

语法：该函数有如下两种表达形式：

- munlock
- munlock(fun)

描述：munlock 命令将对被锁定的 M 文件进行解锁操作，这样可以使用"clear"命令来删除指定的 M 文件。

munlock(fun)命令将对内存中被锁定的 M 文件进行解锁操作。只有当需要对使用 mlock 命令进行锁定的 M 文件解锁时才可以使用 munlock(fun)命令。

7. 打开工作空间内的变量

名称：openvar

在数组编辑器中打开工作空间内的变量。

语法：openvar('name')

描述：openvar('name')命令将在数组编辑器中打开工作空间内的变量以便使用图形化调试。数组必须由数字构成。

举例：

在 MATLAB 的命令窗口中输入：

myArray=rand(5);

openvar('myArray')

MATLAB 将在数组编辑器中打开指定名称为"myArray"的变量。用户可以在编辑器中对该数组进行调试，如图 1-5 所示。

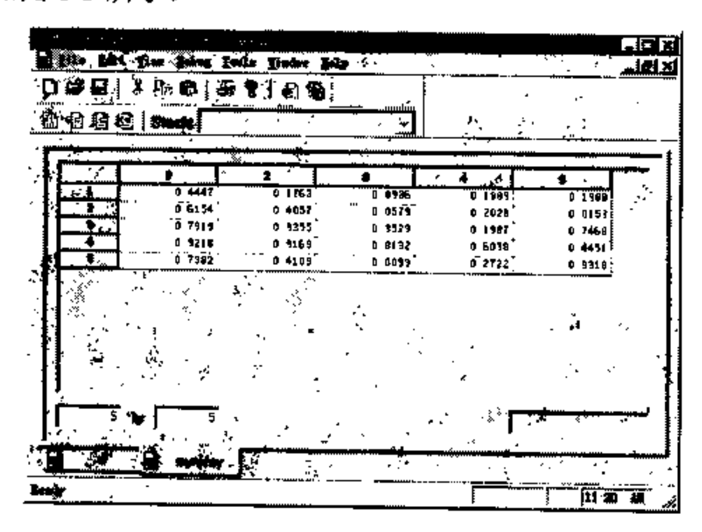


图 1-5　数组编辑器中的变量

8. 合并工作空间的内存

名称：pack

合并工作空间的内存。

语法：该函数有如下两种表达形式：

- pack
- pack filename

描述：pack 命令可以通过对工作空间内的信息进行压缩的方法来释放内存空间。用户必须在拥有写权限的目录中运行 pack 命令。

pack filename 命令将为那些用于保存变量的临时文件添加一个可选的文件名，否则该临时文件将使用“pack.tmp”的文件名。

pack 命令将不会影响已经分配给 MATLAB 进程的内存量。若需要释放这些内存，用户必须退出 MATLAB。由于 MATLAB 使用堆式内存管理，扩展的 MATLAB 会话将产生内存碎片。当内存碎片过多时，尽管有大量的未被使用的内存空间，但可能没有足够多的连续内存来存储一个大型变量。

pack 命令将通过以下方式释放内存：

(1) 将所有变量保存到磁盘上名为“pack.tmp“的临时文件。

(2) 从内存中清除所有的变量及函数。

(3) 从名为“pack.tmp“的临时文件重新载入变量。

(4) 删除名为“pack.tmp“的临时文件。

若使用 pack 命令后仍未有足够内存，用户必须从 MATLAB 工作空间内删除部分变量。

举例：

本例首先改变当前目录到一个拥有写权限的临时目录，接着运行 pack 命令，最后返回到原先的目录。本例中用到的命令如下：

```
mydir=pwd;
cd(tempdir);
pack
cd(mydir)
```

9. 变量保存

名称：save

将工作空间内的变量保存到磁盘上。

语法：该函数有如下几种表达形式：

- save
- save filename
- save filename variables
- save filename options
- save filename variables options

描述：save 命令将把工作空间内的变量用二进制格式保存到磁盘上，并给出一个“matlab.mat”的文件名。

save filename 命令将把工作空间内的变量用二进制格式保存到磁盘上，并给出一个指定文件名来代替默认的“matlab.mat”的文件名。

save filename variables 命令将仅仅把工作空间内的指定变量用二进制格式保存到磁盘上，并给出一个指定的文件名代替默认的“matlab.mat”的文件名。

save filename options 命令将根据所给出的选项把工作空间内的变量用二进制格式保存到磁盘上，并给出一个指定文件名来代替默认的“matlab.mat”的文件名。

save filename variables options 命令将根据所给出的选项把工作空间内的指定变量用二进制格式保存到磁盘上，并给出一个指定文件名来代替默认的“matlab.mat”的文件名。

选项的值见表 1-7。

表 1-7　　参数 options 的值

值	数据存储格式
-ascii	用 8 位 ASCII 格式存储
-ascii -double	用 16 位 ASCII 格式存储
-ascii –tabs	用 8 位 ASCII 格式存储，制表分离
-ascii –double –tabs	用 16 位 ASCII 格式存储，制表分离
-V4	用 MATLAB4 能够载入的格式存储
-append	添加到一个已经存在的指定 MAT 文件中

save 命令可以把 MATLAB 命令存储到磁盘上。save 命令和 load 命令一起使用可以将数字数组作为 ASCII 数据文件输入和输出。MAT 文件是由“save”命令创建并能够被“load”命令读取的二进制双精度 MATLAB 格式的文件。这样就可以在一台计算机上创建并在另外的计算机上读取 MAT 文件。MAT 文件也能够被 MATLAB 以外的其他外部程序操作。

如果用户希望使用“-V4”选项将 MATLAB5 以上版本的数据保存为 MATLAB4 版本能够载入的数据格式，用户必须使用 MATLAB4 版本支持的文件名。另外，用户只能保存 MATLAB4 版本能够理解的数据结构，这就意味着用户将不能保存结构、单元数组、多维数组及对象这些 MATLAB5 版本后出现的数据结构。

此外，当使用“-ascii”选项保存复数数据时，复数的虚部将丢失，这是因为 MATLAB 不能载入非数值数据“i”。

使用“save”命令保存的二进制文件格式将取决于文件中每个数组的大小及类型。表 1-8 列出了当数组元素不同范围内存储每个元素需要的字节数。

表 1-8　　存储每个数组元素需要的字节数

数组元素范围	存储每个元素需要的字节数
0 到 255	1
0 到 65535	2
-32767 到 32767	2
-231+1 到 231+1	4
其他	8

包含 C 和 Fortran 程序的 API 通过外部程序来读写 MAT 文件。这是 MATLAB 推荐的一种访问方法，因为依靠指定文件格式进行访问的方法将有可能被改变。

举例：

若需要保存“myfile”文件中的“a”、“b”、“c”变量，可以在 MATLAB 命令窗口中输入 save myfile a b c。

此外，MATLAB 也接受如下的命令格式：save('myfile','a','b','c')。

该语句的作用也是保存“myfile”文件中的“a”、“b”、“c”变量。

10. 保存图形和模型

名称： saveas

按照指定的格式保存图形和模型。

语法：该函数有如下两种表达形式：

- saveas(h,'filename.ext')
- saveas(h,'filename','format')

描述：saveas(h,'filename.ext')命令将使用带有“h”句柄、名为“filename.ext”的文件保存图形和模型。保存的文件格式取决于文件的扩展名。“ext”的取值见表 1-9。

表 1-9　　参数“ext”的值

ext 的值	文件格式
Ai	Adobe Illustrator '88 文件格式
bmp	Windows 位图文件格式
emf	Enhanced metafile 文件格式
eps	EPS Level 1 文件格式
fig	MATLAB 图形文件格式(对 MATLAB 模型无效)
jpg	JPEG 图像格式
m	MATLAB 的 M 文件格式(对 MATLAB 模型无效)
pbm	可移植位图文件格式
pcx	24 位画笔文件格式
pgm	可移植灰度映射文件格式
png	可移植网络图形文件格式
ppm	可移植像素映射文件格式
tif	TIFFT 图像文件格式

saveas(h,'filename','format')命令将把图形和模型保存到使用指定格式、带有“h”句柄的指定文件中。

format 的有效选项见表 1-10。

表 1-10　　format 的有效选项

format 的有效选项	作用
fig	将图形保存到二进制单精度 FIG 文件中，使用 OPEN 命令重新载入
m	将图形保存到二进制 FIG 文件中，并产生可调用的 M 文件
mfig	将图形保存到二进制 FIG 文件中，并产生可调用的 M 文件，同 m
mmat	将图形保存到带有参数值对的 M 文件中

除了以上选项外，format 的可选项还包括打印许可的图案。

举例：

当用户需要保存图表编辑器中的当前图形时，可以使用 saveas(h,'filename.ext')命令指定文件的扩展名。例如，要将名为“myfig”的图形保存为 JPEG 图像格式，可以在 MATLAB 命令窗口中输入 saveas(gcf,'myfig.jpg')。

若要为该图形指定格式(例如 FIG 格式)，但不指定扩展名，可以在 MATLAB 命令窗口中输入如下命令 saveas(gcf,'myfig','fig')。

11. 数组维数

名称：size

数组维数

语法：该函数有如下几种表达形式：

- d = size(X)

- [m,n] = size(X)
- m = size(X,dim)
- [d1,d2,d3,...,dn] = size(X)

描述：d = size(X)命令将使用带有 N 个元素的向量 d 返回 N 维数组 X 的各维长度。

[m,n] = size(X)命令将使用变量 m 和 n 返回数组 X 的大小。

m = size(X,dim)命令将使用变量 m 返回数组 X 的第“dim”维的长度。

[d1,d2,d3,...,dn] = size(X)命令将使用 d1,d2,d3,...,dn 返回多维数组各维的长度。

举例：

若想用一个向量表示一个四维随机数组的大小，可以在 MATLAB 命令窗口中输入：

```
x=rand(2,4,6,8);
d=size(x)
```

结果为

```
d =
     2     4     6     8
```

若想知道一个三维随机数组的第二维长度，可以在 MATLAB 命令窗口中输入：

```
x=rand(3,6,9);
d=size(x,2)
```

结果为

```
d =   6
```

若要使用不同的变量返回一个三维随机数组各维的长度，可以在 MATLAB 命令窗口中输入：

```
x=rand(2,5,6);
[d1,d2,d3]=size(x)
```

结果为

```
d1 = 2
d2 = 5
d3 = 6
```

12. 变量目录

名称：who，whos

列出内存中的变量目录。

语法：该函数有如下几种表达形式：

- who
- whos
- who global
- whos global
- who -file filename
- whos -file filename
- who ... var1 var2
- whos ... var1 var2

- s = who(...)
- s = whos(...)

描述

who 命令将列出当前内存中所有的变量。

Whos 命令将列出当前的所有变量及其大小和是否有非零的虚部。

who global 命令将列出工作空间内所有全局变量。

whos global 命令将列出工作空间内所有全局变量及其大小和是否有非零的虚部。

who -file filename 命令将列出指定 MAT 文件中的变量。

whos -file filename 将列出指定 MAT 文件中的所有变量及其大小和是否有非零的虚部。

who ... var1 var2 命令将只列出指定的变量。

whos ... var1 var2 命令将只列出指定的变量及其大小和是否有非零的虚部。

s = who(...)命令将返回一个包含有工作空间或指定文件中的变量的名字的单元数组。

s = whos(...)命令将返回一个带有三个字段的结构。该结构的字段名见表 1-11。

表 1-11　　s = whos(...)命令返回的结构

字段名	解释
name	变量名
bytes	分配给该数组的字节数
class	变量的类

13. 工作空间浏览器

名称：workspace

显示工作空间浏览器。

语法：workspace

描述：workspace 打开工作空间浏览器。工作空间浏览器是一个用于管理工作空间内容的图形化用户界面。该浏览器提供了一个使用“whos”命令显示的内容的图形化表示。

在 Windows 操作系统中，还可以通过在文件菜单中选择“Show Workspace”命令，或在工具栏中单击“Workspace Browser”按钮打开工作空间浏览器。在 MATLAB 命令窗口中输入 workspace，将打开工作空间浏览器，如图 1-6 所示。

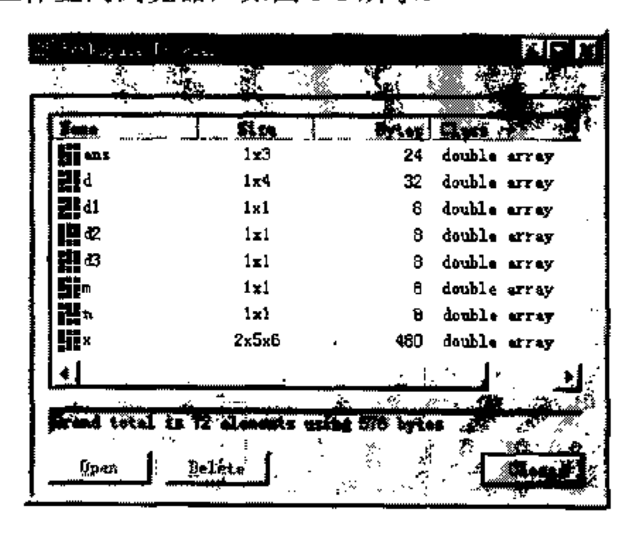


图 1-6　MATLAB 工作空间浏览器

1.3　控制命令窗口

1. 清空命令窗口

名称：clc

清空命令窗口。

语法：clc

描述：clc 命令将清空命令窗口。

2. 禁止或允许显示执行过程

名称：echo

在 M 文件执行时禁止或允许显示执行过程。

语法：该函数有如下几种表达形式：

- echo on
- echo off
- echo fcnname on
- echo fcnname off
- echo fcnname
- echo on all
- echo off all

描述：echo 命令控制 M 文件执行时的显示。正常情况下，执行 M 文件的过程将不显示在屏幕上。在调试和演示时允许 M 文件执行时显示在屏幕上是非常有用的。分别对脚本文件和函数文件执行 echo 命令，其作用稍有不同。对脚本文件，使用 echo 命令较为简单，echo 命令只有两种状态，“on”或“off”。此时的 echo 命令见表 1-12。

表 1-12　　echo 命令

echo 命令	作用
echo on	在所有的脚本文件中打开 echo 命令
echo off	在所有的脚本文件中关闭 echo 命令
echo	切换 echo 命令的状态

对函数文件，使用 echo 命令就显得复杂一些。如果对于一个函数文件，echo 命令是打开的，该文件将被解释执行而不是编译执行。在每个输入行被执行时，该行在屏幕上被显示出来。但这样做的结果将导致执行效率低下，所以一般只在调试时打开 echo 命令。

在函数文件中的 echo 命令见表 1-13。

表 1-13　　在函数文件中的 echo 命令

echo 命令	作用
echo fcnname on	对指定的函数打开 echo 命令
echo fcnname off	对指定的函数关闭 echo 命令
echo fcnname	对指定的函数切换 echo 命令的状态
echo on all	对所有的函数打开 echo 命令
echo off all	对所有的函数关闭 echo 命令

3. 控制输出显示格式

名称：format

控制输出显示格式。

语法：该函数有如下两种表达形式：

- format
- format type

描述：MATLAB 使用双精度来完成所有计算。format 命令仅控制在屏幕上的显示格式。format 命令见表 1-14。

表 1-14　　format 命令

format 命令	说明
format	默认值，同 format short
format short	5 位定点数
format long	15 位定点数
format short e	5 位浮点数
format long e	15 位浮点数
format short g	5 位定点或浮点数
format long g	15 位定点或浮点数
format hex	16 进制数
format bank	货币格式数
format rat	有理数
format +	+、-或空格
format compact	压缩格式
format loose	稀疏格式

举例：

在 MATLAB 命令窗口中输入：

```
a = pi;
format short
a
```

结果为：

```
a =
     3.1416
```

继续输入：

```
format long
a
```

结果为

```
a =
   3.14159265358979
```

继续输入：

```
format short e
a
```

结果为：

```
a =
```

3.1416e+000

继续输入：

format long e

a

结果为：

a =

3.141592653589793e+000

继续输入：

format hex

a

结果为：

a =

400921fb54442d18

继续输入：

format rat

a

结果为：

a =

355/113

继续输入：

format +

a

结果为：

a =

+

4. 光标移动到 MATLAB 命令窗口的初始位置

名称：home

将光标移动到 MATLAB 命令窗口的初始位置。

语法：home

描述：home 命令将把光标移动到 MATLAB 命令窗口的左上角。

举例：

下面的例子将在 MATLAB 的命令窗口的相同位置按照顺序显示 16 个 4*4 随机矩阵。

```
for i =1:16
    home
    A = rand(4)
end
```

5. 控制 MATLAB 命令窗口中的页输出

名称：more

控制 MATLAB 命令窗口中的页输出。

语法：该函数有如下几种表达形式：

- more off
- more on
- more(n)

描述：more off 命令将不允许在 MATLAB 的命令窗口中进行页面调度。

more on 命令将允许在 MATLAB 的命令窗口中进行页面调度。

more(n)命令将控制每页的行数。

默认情况下，该命令的状态为 more off，当将该命令的状态改为 more on 时，每页的默认显示行数为 23。

1.4 使用文件和工作环境

1. 改变工作目录

名称：cd

改变工作目录。

语法：该函数有如下几种表达形式：

- cd
- cd directory
- cd ..

描述：cd 命令将打印出当前目录。

cd directory 命令将把指定目录设置为当前目录。

cd ..命令将把当前目录的上一级目录设置为当前目录。

举例：

下面举出了在不同的操作系统中的 cd 命令的用法。

UNIX: cd /disk1/matlab/toolbox/demos

WINDOWS: cd C:\MATLAB\DEMOS

VMS: cd DISK1:[MATLAB.DEMOS]

2. 复制文件

名称：copyfile

复制文件。

语法：该函数有如下几种表达形式：

- copyfile('source','dest')
- copyfile('source','dest','writable')
- status = copyfile('source','dest')
- [status,msg] = copyfile('source','dest')

描述：copyfile('source','dest')命令将把指定源目录中的文件复制到指定的目的目录中。其中，源目录和目的目录既可以是绝对路径，也可以是相对路径。

copyfile('source','dest','writable')命令将把指定源目录中的文件复制到指定的目的目录

中，并将确认目的目录是可写的。

status = copyfile('source','dest')返回复制文件命令执行后的状态。该状态的值见表 1-15。

表 1-15 复制文件命令执行后的状态的值

值	作用
0	复制文件成功
1	复制文件失败

[status,msg] = copyfile('source','dest')命令将返回复制文件命令执行后的状态，并当发生错误时返回一个非空的错误信息。

3. 删除文件和图形对象

名称：delete

删除文件和图形对象。

语法：该函数有如下两种表达形式：

- delete filename
- delete(h)

描述：delete filename 命令将删除指定的文件。

delete(h)命令将删除带有指定句柄的图形对象。该命令删除对象时不需要确认，即使该对象是一个窗口也可以直接删除。

4. 保存会话

名称：diary

将会话保存到一个磁盘文件中。

语法：该函数有如下几种表达形式：

- diary
- diary filename
- diary off
- diary on

描述：diary 命令将创建一个用户键盘输入并由系统应答的日志文件。使用该命令将输出一个包含报告和其他文档的能够打印的 ASCII 文件。

diary filename 使用指定的文件名，创建一个包含所有的用户键盘输入和大部分输出结果的日志文件。如果该文件已经存在，将自动把输出结果直接添加到该日志文件的结尾处。

diary off 命令将暂停执行 diary 命令。

diary on 命令将恢复执行 diary 命令并使用当前的文件名，如果没有指定日志文件的名称，系统将使用默认的日志文件名“diary”

5. 目录列表

名称：dir

目录列表。

语法：该函数有如下几种表达形式：

- dir
- dir dirname
- names = dir

- names = dir('dirname')

描述：dir 命令将列出当前目录中的所有文件。

dir dirname 命令将列出指定目录中的所有文件。

names = dir 把当前目录中的所有文件返回到一个结构数组中。该数组的字段见表 1-16。

表 1-16　　结构数组的字段名

字段名	解释
name	文件名
date	修改数据
bytes	该文件占用的字节数
isdir	是否为目录名(若是目录，该值为 1；否则为 0)

names = dir('dirname') 命令将把指定目录中的所有文件返回到一个结构数组中。

举例：

若要显示\matlab\work 目录中的文件，可以在 MATLAB 命令窗口中输入：

```
cd matlab\work
dir
```

结果显示为

```
.          ..          myfig.jpg
```

若继续输入

```
names=dir
```

结果显示为

```
names =
3x1 struct array with fields:
    name
    date
    bytes
    isdir
```

6. 编辑 M 文件

名称：edit

编辑一个 M 文件。

语法：该函数有如下几种表达形式：

- edit
- edit fun
- edit file.ext
- edit class/fun
- edit private/fun
- edit class/private/fun

描述：edit 命令将打开一个新的编辑器窗口。

edit fun 命令将使用默认的编辑器打开指定的 M 文件。

edit file.ext 命令将使用默认的编辑器打开指定的文本文件。

edit class/fun 命令将使用默认的编辑器打开指定的方法文件。

edit private/fun 命令将使用默认的编辑器打开指定的私有函数文件。

edit class/private/fun 命令将使用默认的编辑器打开指定的私有方法文件。

使用 Windows 操作系统的用户可从“File”菜单中选择“New”或“Open”命令来启动 MATLAB 编辑器；在工具栏中单击“New”或“Open”按钮也可以打开 MATLAB 编辑器。

使用 UNIX 操作系统的用户在安装 MATLAB 时，可以为 MATLAB 指定默认的编辑器。如果需要改变设置，可以编辑用户的~home/.Xdefaults 文件。如果 MATLAB 自带的编辑器被设置为系统默认编辑器，请在~home/.Xdefaults 文件中关闭之。

用户可以在~home/.Xdefaults 文件中进行如下操作：

matlab*builtInEditor: Off

matlab*graphicalDebugger: Off

然后在启动 MATLAB 前，运行

xrdb -merge ~home/.Xdefaults

如果用户的编辑器本身设置为“Off”，那么直接使用选项

matlab*externalEditorCommand: $EDITOR $FILE &

控制编辑命令的用法。MATLAB 将替换选项“$EDITOR”指定的编辑器名和选项“$FILE”指定的文件名。

7. 文件名的各个部分

名称：fileparts

返回文件名的各个部分。

语法：[path,name,ext,ver] = fileparts(file)

描述：[path,name,ext,ver]=fileparts(file)返回指定文件的路径、文件名、扩展名和版本。

8. 建立全文件名

名称：fullfile

使用指定部分建立全文件名。

语法：fullfile(dir1,dir2, ...,filename)

描述：fullfile(dir1,dir2, ...,filename)命令将根据指定的文件名和目录建立一个全文件名。

举例：

在 MATLAB 命令窗口中输入：

fullfile(matlabroot,'toolbox','local','cedit.m')

结果为

ans =

D:\MATLAB\toolbox\local\cedit.m

上面的语句不但在 Windows 系统中使用，在 UNIX 系统中也能产生同样的作用。但命令 fullfile(matlabroot,'toolbox/local/cedit.m')只能在 UNIX 系统中使用。

9. 内存中的函数

名称：inmem

内存中的函数。

语法：该函数有如下几种表达形式：

- M = inmem

- [M,X] = inmem

描述：M = inmem 命令将返回一个包含内存中所有 M 文件名的字符串单元数组。

[M,X] = inmem 命令除了返回一个包含内存中所有 M 文件名的字符串单元数组以外，还将返回一个包含内存中所有 MEX 文件名的单元数组。

举例：

下面的命令将返回执行函数“erfc”时所需要调用的 M 文件。

```
clear all;
erfc(0.8);
M = inmem
```

结果为

```
M =
    'repmat'
    'erfcore'
    'erfc'
```

10. 在 UNIX 系统中列出目录

名称：ls

在 UNIX 系统中列出目录。

语法：ls

描述：ls 命令将列出 UNIX 系统中的目录。

11. 返回安装 MATLAB 的根目录

名称：matlabroot

返回安装 MATLAB 的根目录。

语法：rd = matlabroot

描述：rd = matlabroot 命令将使用变量“rd”返回安装 MATLAB 的根目录。

举例：

例如，在 MATLAB 命令窗口中输入 rd = matlabroot，

将返回安装 MATLAB 的根目录。

```
rd =
D:\MATLAB
```

12. 新建目录

名称：mkdir

新建目录。

语法：该函数有如下几种表达形式：

- mkdir('dirname')
- mkdir('parentdir','newdir')
- status = mkdir('parentdir','newdir')
- [status,msg] = mkdir('parentdir','newdir')

描述：mkdir('dirname')命令将在当前目录下使用指定目录名新建目录。

mkdir('parentdir','newdir')命令将在指定父目录下使用指定目录名新建目录。

status = mkdir('parentdir','newdir')命令将返回执行在指定父目录下使用指定目录名新建目录操作后的状态。该状态的值见表 1-17。

表 1-17 新建目录操作后的状态的值

值	作用
0	未能创建新目录
1	新目录被成功创建
2	该目录已经存在

[status,msg] = mkdir('parentdir','newdir')命令将返回执行在指定父目录下使用指定目录名新建目录操作后的状态，并在发生错误时使用“msg”返回一个非空的错误信息。

13. 打开文件

名称：open

基于扩展名打开文件。

语法：open('name')

描述：open('name')命令将根据指定的文件类型打开文件。“name”的类型见表 1-18。

表 1-18 参数“name”的类型

name 的类型	解释
变量	在数组编辑器中打开该变量
.fig 文件	使用图形窗口打开该文件
.m 文件	在编辑器中打开 M 文件
.p 文件	在编辑器中打开对应的 M 文件
.mdl 文件	在 SIMULINK 中打开指定的模型
其他文件	打开用户定义的文件

使用 open('name')命令打开文件时，若没有指定扩展名，open 命令将打开由 which(name)命令返回的对象，该对象有可能是一个变量、函数或模型。如果存在一个不带扩展名的匹配文件名，open 命令将在编辑器中打开该文件；如果没有找到匹配的文件，open 命令将在路径中搜索同名的 M 文件，找到后，将在编辑器中打开该 M 文件。

当 open 命令打开的是一个变量时，该命令将调用“openvar”函数。

举例：

若使用 open 命令打开名为“mydata”、不带扩展名的文件，在 MATLAB 命令窗口中输入 open('mydata')。

如果“mydata”文件存在，则在编辑器中打开该文件；如果“mydata”文件不存在，则在编辑器中打开“mydata.m”文件。

在上例中，如果搜索路径中存在同名但有不同扩展名的两个文件“mydata.m”和“mydata.mdl”，命令 open('mydata')将在 SIMULINK 中打开“mydata.mdl”文件，这是因为模型文件的优先级比 M 文件的优先级高。

14. 显示当前目录

名称：pwd

显示当前目录。

语法：s = pwd

描述：s = pwd 命令用变量“s”返回当前目录。

15. 返回系统的临时目录的名称

名称：tempdir

返回系统的临时目录的名称。

语法：tmp_dir = tempdir

描述：当系统有临时目录时，tmp_dir=tempdirl 使用变量“tmp_dir”返回临时目录名称。

举例：

在 MATLAB 命令窗口中输入 tempdir，

结果为

ans =

C:\WINDOWS\TEMP\

16. 临时文件名

名称：tempname

给临时文件一个唯一的文件名。

语法：tempname

描述：tempname 命令将返回一个以“tp”开头的唯一字符串。

举例：

在 MATLAB 命令窗口中输入 tempname，

结果为

ans =

C:\WINDOWS\TEMP\tp400298

17. 指出感叹号后的内容是操作系统命令

名称：！

用于指出感叹号后的内容是操作系统命令。

举例：

若需要在 MATLAB 中调用 Windows 系统的磁盘扫描程序，可以在 MATLAB 命令窗口中输入!scandisk，此时将直接打开 Windows 系统的磁盘扫描程序。

1.5 启动和退出 MATLAB

1. 启动 M 文件

名称：matlabrc

MATLAB 的启动 M 文件。

语法：matlabrc

描述：在启动 MATLAB 时，MATLAB 将自动执行主 M 文件“matlabrc.m”。

启动 MATLAB 时，若存在“startup.m”文件，“matlabrc.m”将调用“startup.m”文件。在多用户系统或网络环境中，只有系统管理员才能够调用“matlabrc.m”文件。在单机环境中，用户可以在 MATLAB 搜索目录中创建一个“startup.m”文件，通过“startup.m”文件，用户可以定义物理常数、工程转换系数、默认图形及任何可以在工作空间内预定义的内容。

举例：

若用户希望让图形窗口中的工具栏一直保持打开的状态，可以将“matlabrc.m”文件中的语句%set(0,'defaultfiguretoolbar','figure')前的注释符号删除。

2. 退出 MATLAB

名称： quit

退出 MATLAB。

语法： 该函数有如下几种表达形式：

- quit
- quit cancel
- quit force

描述： quit 命令将退出 MATLAB 系统。若存在“finish.m”文件，在退出 MATLAB 前，将运行“finish.m”文件。用户可以使用“finish.m”文件保存工作空间内的变量或执行其他操作。如果“finish.m”文件运行时发生错误，quit 命令将被取消。

quit cancel 命令将执行“finish.m”文件，并取消 quit 命令。

quit force 命令将强行退出 MATLAB，并绕开“finish.m”文件。这是因为当“finish.m”文件中存在错误而不允许执行退出命令时，使用该命令将忽略“finish.m”文件以便正常退出 MATLAB。

举例：

在 MATLAB 中存在两个“finish.m”文件的例子。用户可以使用这两个例子来创建自己的“finish.m”文件。MATLAB 中的两个“finish.m”文件例子分别是“finishsav.m”和“finishdlg.m”。

其中，“finishsav.m”文件的作用是当退出 MATLAB 系统时，保存工作空间内的所有内容到一个 MAT 文件中。该文件的代码如下：

```
% FINISHSAV    Save workspace variables
%     Change the name of this file to FINISH.M
%     and put it anywhere on your MATLAB path.
%     When you quit MATLAB this file will be executed.
%     This script saves all the variables in the
%     work space to a MAT-file.

%     Copyright (c) 1984-98 by The MathWorks, Inc.
%     $Revision: 1.1 $   $Date: 1998/05/15 20:51:03 $

disp('Saving workspace data');
save
```

“finishdlg.m”文件的作用是显示一个对话框，用户可以在对话框中选择是否取消执行 quit 命令。该文件的代码如下：

```
%FINISHDLG    Display a dialog to cancel quitting
%     Change the name of this file to FINISH.M and
```

```
%    put it anywhere on your MATLAB path. When you
%    quit MATLAB this file will be executed.

%    Copyright (c) 1984-98 by The MathWorks, Inc.
%    $Revision: 1.3 $   $Date: 1998/06/09 21:55:43 $

button = questdlg('Ready to quit?', ...
                  'Exit Dialog','Yes','No','No');
switch button
  case 'Yes',
    disp('Exiting MATLAB');
      %Save variables to matlab.mat
      save
  case 'No',
    quit cancel;
end
```

3. 运行 MATLAB 启动文件

名称：startup

运行 MATLAB 启动文件。

语法：startup

描述：启动 MATLAB 时，若存在"startup.m"文件，"matlabrc.m"文件将调用"startup.m"文件。

用户可以在 MATLAB 搜索目录中创建一个"startup.m"文件，通过"startup.m"文件，用户可以定义物理常数、工程转换系数、默认图形及任何可以在工作空间内预定义的内容。

第二章　运算符和逻辑函数

2.1　算术运算符

1．加法

名称：+

加法。

语法：A+B

描述：A+B 命令将对矩阵 A 和 B 进行加法运算，结果是一个由矩阵 A 和 B 相应元素的和组成的新矩阵。

A 和 B 必须为同维矩阵，或其中之一为一个标量。当 A 和 B 其中之一为标量时，结果是矩阵中的每个元素与该标量做加法运算得出的新矩阵。

举例：

在 MATLAB 命令窗口中输入：

```
A = [1 2;3 4;5 6];
B = 3;
A + B
```

结果为：

```
ans =
     4     5
     6     7
     8     9
```

继续输入：

```
C = [6 5;4 3;2 1];
A + C
```

结果为：

```
ans =
     7     7
     7     7
     7     7
```

2．减法

名称：-

减法。

语法：A-B

描述：A-B 命令将对矩阵 A 和 B 进行减法运算，结果是一个由矩阵 A 和 B 相应元素的差组成的新矩阵。

A 和 B 必须为同维矩阵，或其中之一为一个标量。当 A 和 B 其中之一为标量时，结果是矩阵中的每个元素与该标量做减法运算得出的新矩阵。

举例：

在 MATLAB 命令窗口中输入：

```
A = magic(4);
B = 2;
A - B
```

结果为：

```
ans =
    14     0     1    11
     3     9     8     6
     7     5     4    10
     2    12    13    -1
```

```
C = ones(4);
A - C
```

结果为：

```
ans =
    15     1     2    12
     4    10     9     7
     8     6     5    11
     3    13    14     0
```

3. 矩阵乘法

名称：*

矩阵乘法。

语法：A*B

描述：A*B 命令将对矩阵 A 和 B 进行乘法运算。

若 A、B 全为矩阵，则 A 矩阵的列数必须等于 B 矩阵的行数；若其中之一为标量，结果是矩阵中的每个元素与该标量做乘法运算得出的新矩阵。

举例：

在 MATLAB 命令窗口中输入：

```
A = magic(3);
B = 6;
A * B
```

结果为：

```
ans =
    48     6    36
    18    30    42
```

```
    24    54    12
```

继续输入：

```
C = [1 1;2 2;3 3];
A * C
```

结果为：

```
ans =
    28    28
    34    34
    28    28
```

4. 数组乘法

名称：.*

数组乘法。

语法：A.*B

描述：A.*B 命令将对数组 A 和 B 进行乘法运算。A 和 B 必须同维，或其中之一为标量。

举例：

在 MATLAB 命令窗口中输入：

```
A = [1 2 3;4 5 6;7 8 9];
B = magic(3);
A .* B
```

结果为：

```
ans =
     8     2    18
    12    25    42
    28    72    18
```

5. 矩阵乘方

名称：^

矩阵乘方。

语法：A^B

描述：当 A 为方阵，而 B 为一个大于 1 的整数时，A^B 的结果是 A 的 B 次幂，由 A 自乘 B 次得到；若 B 为其他数值，A^B 的结果为各特征值和特征向量的乘方。

当 B 为方阵，而 A 为一个数时，A^B 的结果是 A 的 B 次幂，由各特征值和特征向量的乘方得到。

若 A、B 全为矩阵，A^B 将返回错误。

举例：

在 MATLAB 命令窗口中输入：

```
A = magic(3);
B = 2;
A ^ B
```

结果为：

```
ans =
     91    67    67
     67    91    67
     67    67    91
```

继续输入：

```
C = 1.6;
A ^ C
```

结果为：

```
ans =
  33.6231 - 0.3282i  19.0291 - 2.9204i  23.5111 + 3.2485i
  20.8219 - 0.4528i  30.9339 - 4.0295i  24.4075 + 4.4823i
  21.7183 + 0.7810i  26.2003 + 6.9499i  28.2448 - 7.7309I
```

继续输入：

```
B ^ A
```

结果为：

```
ans =
  1.0e+004 *
    1.0942    1.0906    1.0921
    1.0912    1.0933    1.0924
    1.0915    1.0930    1.0923
```

6. 数组乘方

名称：.^

数组乘方。

语法：A.^B

描述：A.^B 命令将分别以 A 中的元素为底，B 中的相应元素为幂做乘方运算。A 和 B 必须为同维数组，或其中之一为一个标量。

举例：

在 MATLAB 命令窗口中输入：

```
A = [1 2 3;4 5 6;7 8 9];
B = 3;
A .^ B
```

结果为

```
ans =
     1      8     27
    64    125    216
   343    512    729
```

继续输入：

```
B .^ A
```

结果为：

```
ans =
          3          9         27
         81        243        729
       2187       6561      19683
```

继续输入：

```
C = 2*ones(3);
A .^ C
```

结果为：

```
ans =
     1     4     9
    16    25    36
    49    64    81
```

7. 矩阵左除

名称： \

矩阵左除。

语法： A\B

描述： A\B 命令为矩阵 A 左除矩阵 B。

若 A 为方阵，A\B 的结果与 INV(A)*B 的结果相同；若 A 为 N×N 方阵，B 为有 N 个元素的列向量或含有若干这样的列的矩阵，则 X=A\B 是方程 A*X=B 的高斯消元法的解。

若 A 为 M×N 矩阵，B 是含有 M 个元素的列向量或含有若干这样的列的矩阵，则 X=A\B 是欠定或超定方程 A*X=B 最小二乘意义下的解。A 的有效秩 k 是用带旋转的 QR 分解得到的，至多在每列 k 个非零元素上求解 X。

8. 矩阵右除

名称： /

矩阵右除。

语法： B/A

描述： B/A 命令为矩阵 A 右除矩阵 B。其结果大致与 B*INV(A)相同，更准确地，B/A=(A'\B')'。

9. 数组左除

名称： .\

数组左除。

语法： A.\B

描述： A.\B 命令将得到一个矩阵，该矩阵的元素为数组 A 和数组 B 中的每个相应元素进行 B(i,j)/A(i,j)运算的结果。A 和 B 必须为同维数组，或其中之一为标量。

10. 数组右除

名称： ./

数组右除。

语法： A./B

描述： A./B 命令将得到一个矩阵，该矩阵的元素为数组 A 和数组 B 中的每个相应元素

进行 A(i,j)/B(i,j)运算的结果。A 和 B 必须为同维数组，或其中之一为标量。

11. 克罗内克张量积

名称：kron

克罗内克张量积。

语法：K = kron(X,Y)

描述：K = kron(X,Y)命令将给出矩阵 X 和矩阵 Y 的克罗内克张量积。结果将得到一个大矩阵，取 X 和 Y 的元素间所有可能的乘积。

举例：

在 MATLAB 命令窗口中输入：

```
X = [1 2;3 4;5 6];
Y = [2 4;6 8];
kron(X,Y)
```

结果为：

```
ans =
     2     4     4     8
     6     8    12    16
     6    12     8    16
    18    24    24    32
    10    20    12    24
    30    40    36    48
```

2.2 关系运算符

1. 小于

名称：<

小于。

语法：A < B

描述：A < B 命令将对 A 和 B 的相应元素进行关系运算，并返回一个与 A 和 B 维数相同的数组。当 A 和 B 的相应元素进行关系运算的结果为真时(即 A 中的元素小于 B 中的相应元素)，在新数组的相同位置返回一个值为 1 的元素；否则在新数组的相同位置返回一个值为 0 的元素。A 和 B 其中之一可以为标量。

举例：

在 MATLAB 命令窗口中输入：

```
A = magic(3);
B = [1 2 3;4 5 6;7 8 9];
A < B
```

结果为

```
ans =
```

```
     0     1     0
     1     0     0
     1     0     1
```

2. 大于

名称：>

大于。

语法：A > B

描述：A > B 命令将对 A 和 B 的相应元素进行关系运算，并返回一个与 A 和 B 维数相同的数组。当 A 和 B 的相应元素进行关系运算的结果为真时(即 A 中的元素大于 B 中的相应元素)，在新数组的相同位置返回一个值为 1 的元素；否则在新数组的相同位置返回一个值为 0 的元素。A 和 B 其中之一可以为标量。

举例：

在 MATLAB 命令窗口中输入：

```
A = magic(3);
B = [1 2 3;4 5 6;7 8 9];
A > B
```

结果为：

```
ans =
     1     0     1
     0     0     1
     0     1     0
```

3. 小于等于

名称：<=

小于等于。

语法：A <= B

描述：A <= B 命令将对 A 和 B 的相应元素进行关系运算，并返回一个与 A 和 B 维数相同的数组。当 A 和 B 的相应元素进行关系运算的结果为真时(即 A 中的元素小于等于 B 中的相应元素)，在新数组的相同位置返回一个值为 1 的元素；否则在新数组的相同位置返回一个值为 0 的元素。A 和 B 其中之一可以为标量。

举例：

在 MATLAB 命令窗口中输入：

```
A = magic(3);
B = 6;
A <= B
```

结果为：

```
ans =
     0     1     1
     1     1     0
     1     0     1
```

4. 大于等于

名称：>=

大于等于。

语法：A >= B

描述：A >= B 命令将对 A 和 B 的相应元素进行关系运算，并返回一个与 A 和 B 维数相同的数组。当 A 和 B 的相应元素进行关系运算的结果为真时(即 A 中的元素大于等于 B 中的相应元素)，在新数组的相同位置返回一个值为 1 的元素；否则在新数组的相同位置返回一个值为 0 的元素。A 和 B 其中之一可以为标量。

举例：

在 MATLAB 命令窗口中输入：

```
A = magic(3);
B = 6;
A >= B
```

结果为：

```
ans =
     1     0     1
     0     0     1
     0     1     0
```

5. 等于

名称：==

等于。

语法：A == B

描述：A == B 命令将对 A 和 B 的相应元素进行关系运算，并返回一个与 A 和 B 维数相同的数组。当 A 和 B 的相应元素进行关系运算的结果为真时(即 A 中的元素等于 B 中的相应元素)，在新数组的相同位置返回一个值为 1 的元素；否则在新数组的相同位置返回一个值为 0 的元素。A 和 B 其中之一可以为标量。

举例：

在 MATLAB 命令窗口中输入：

```
A = magic(3);
B = [1 2 3;4 5 6;7 8 9];
A == B
```

结果为：

```
ans =
     0     0     0
     0     1     0
     0     0     0
```

6. 不等于

名称：~=

不等于。

语法： A ~= B

描述： A ~= B 命令将对 A 和 B 的相应元素进行关系运算，并返回一个与 A 和 B 维数相同的数组。当 A 和 B 的相应元素进行关系运算的结果为真时(即 A 中的元素不等于 B 中的相应元素)，在新数组的相同位置返回一个值为 1 的元素；否则在新数组的相同位置返回一个值为 0 的元素。A 和 B 其中之一可以为标量。

举例：

在 MATLAB 命令窗口中输入：

```
A = magic(3);
B = [1 2 3;4 5 6;7 8 9];
A ~= B
```

结果为：

```
ans =
     1     1     1
     1     0     1
     1     1     1
```

2.3 逻辑运算符

1. 逻辑与

名称： &

逻辑与。

语法： A&B

描述： A&B 命令将对 A 和 B 的相应元素进行逻辑与运算，并返回一个与 A 和 B 维数相同的数组。当 A 中的元素和 B 中的相应元素均为非零元素时，在新数组的相同位置返回一个值为 1 的元素；否则在新数组的相同位置返回一个值为 0 的元素。

A 和 B 其中之一可以为标量。A&B 命令在 MATLAB 中还可以表示为 and(A,B)。

举例：

在 MATLAB 命令窗口中输入：

```
A = [0 1 0;1 0 1;0 0 0];
B = [0 0 0;1 1 1;1 0 1];
A & B
```

结果为：

```
ans =
     0     0     0
     1     0     1
     0     0     0
```

2. 逻辑或

名称： |

逻辑或。

语法：A|B

描述：A|B 命令将对 A 和 B 的相应元素进行逻辑或运算，并返回一个与 A 和 B 维数相同的数组。当 A 中的元素和 B 中的相应元素中的任一元素为非零元素时，在新数组的相同位置返回一个值为 1 的元素；否则在新数组的相同位置返回一个值为 0 的元素。

A 和 B 其中之一可以为标量。A|B 命令在 MATLAB 中还可以表示为 or(A,B)。

举例：

在 MATLAB 命令窗口中输入：

```
A = [0 1 0;1 0 1;0 0 0];
B = [0 0 0;1 1 1;1 0 1];
A | B
```

结果为：

```
ans =
     0     1     0
     1     1     1
     1     0     1
```

3. 逻辑非

名称：~

逻辑非。

语法：~A

描述：~A 命令将对 A 中的元素进行逻辑非运算。当 A 中的元素为非零元素时，逻辑非运算将返回 0；当 A 中的元素为 0 时，逻辑非运算将返回 1。

~A 命令在 MATLAB 中还可以表示为 not(A)。

举例：

在 MATLAB 命令窗口中输入：

```
A = [-1 0 1;4 0 -4;9 0 9];
~A
```

结果为：

```
ans =
     0     1     0
     0     1     0
     0     1     0
```

4. 逻辑异或

名称：xor

逻辑异或。

语法：xor(A,B)

描述：xor(A,B)命令将对 A 和 B 的相应元素进行逻辑异或运算，并返回一个与 A 和 B 维数相同的数组。当 A 中的元素和 B 中的相应元素一为非零元素、一为 0 时，在新数组的相同位置返回一个值为 1 的元素；否则在新数组的相同位置返回一个值为 0 的元素。

A 和 B 其中之一可以为标量。

举例：

在 MATLAB 命令窗口中输入：

```
A = [-1 0 1;4 0 -4;9 0 9];
B = [0 0 0;1 1 1;0 1 0];
xor(A,B)
```

结果为：

```
ans =
     1     0     1
     0     1     0
     1     1     1
```

继续输入：

```
C = 1;
xor(A,C)
```

结果为：

```
ans =
     0     1     0
     0     1     0
     0     1     0
```

2.4　特殊运算符

1. 冒号

名称：:

冒号。

描述：冒号是 MATLAB 中最有用的运算符之一。冒号可以用于创建向量和下标数组。

当冒号用于创建向量时：

m:n 表示向量[m,m+1,m+2,…,n]；当 m>n 时，该语句将产生一个空向量。

m:k:n 表示向量[m,m+k,m+2k, …,m+l*k]，其中 l=fix(n-m)/k；当 k>0 且 m>n 时，或者 k<0 且 m<n 时，该语句将产生一个空向量。

当冒号用于创建下标数组时，可以挑选出指定的行、列及矩阵。

A(:,j)将挑选出 A 中的第 j 列；

A(i,:)将挑选出 A 中的第 i 行；

A(j:k)将挑选出 A(j)，A(j+1)，…，A(k)；

A(:,j:k)将挑选出 A(:,j)，A(:,j+1)，…，A(:,k)；

A(:,:,k)将挑选出三维数组 A 中的第 k 页。

举例：

在 MATLAB 命令窗口中输入：

```
2:8
```

结果为：

```
ans =
     2     3     4     5     6     7     8
```

继续输入：

```
2:.5:6
```

结果为：

```
ans =
  Columns 1 through 7
    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
  Columns 8 through 9
    5.5000    6.0000
```

在 MATLAB 命令窗口中输入：

```
A = magic(3);
A(:,3)
```

结果为：

```
ans =
     6
     7
     2
```

继续输入：

```
A(2,:)
```

结果为：

```
ans =
     3     5     7
```

继续输入：

```
A(2,3)
```

结果为：

```
ans =
     7
```

2. *方括号*

名称：[ ]

方括号

描述：方括号用于构成向量和矩阵。向量和矩阵可以用于方括号内。类似[A B;C]的格式是被允许的，条件是 A 的行数等于 B 的行数，且 A 的列数与 B 的列数之和等于 C 的列数。

举例：

例如，[1 2 3]是一个包含三个元素的行向量，元素间用空格隔开；[1,2,3]是一个同样的行向量，区别仅在于元素间用逗号隔开。[1;2;3]是一个包含三个元素的列向量。元素间使用分号隔开。

[1 2 3;1 2 3;1 2 3]是一个三行、三列的矩阵，共有9个元素，元素间使用空格隔开，行间使用分号隔开。

3. 圆括号

名称：( )

圆括号。

描述：圆括号通常被用于表示算术表达式的优先级，还可以在圆括号中放置函数参数，此外，圆括号还用于放置向量和矩阵的下标。

举例：

例如，对于矩阵：

```
A = magic(4)
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

输入命令：

```
A(8)
```

结果将显示矩阵A中的第八个元素，

```
ans =
    14
```

继续输入：

```
A([2,3,4])
```

结果将分别显示矩阵A中的第二、三、四个元素。

```
ans =
     5     9     4
```

4. 大括号

名称：{ }

大括号。

描述：大括号通常被用于单元数组分配语句中。

5. 句点

名称：.

句点。

描述：在MATLAB中，句点有三个作用：(1)可以用作十进制的小数点；(2)可以用于表示数组运算；(3)可以用于字段访问。

举例：

当句点出现在数字中时，表示十进制的小数点，如3.14。

当句点出现在运算符“* / \ ^ '”前时，表示数组运算，如A.*B。

当句点出现在结构中时，表示用于字段访问，如A.(field)。

6. 父目录

名称：..

父目录。

描述：

“..”表示当前目录的父目录。该符号通常与命令 CD 和 CHDIR 一起使用。

7. 连续点

名称：...

连续点。

描述：把连续点放于一行的末尾，表示该行并未写完，接下来的一行是上一行的继续。

举例：

在 MATLAB 命令窗口中输入：

```
a = 1 + 1/2 + 1/4 +1/16 + 1/32 + 1/64 + 1/128 ...
     +1/256 + 1/512 + 1/1024;
```

8. 逗号

名称：,

逗号。

描述：逗号常用于隔开矩阵下标和函数的参数，也可以用于多语句行中分开各语句。

9. 分号

名称：;

分号。

描述：分号常用于方括号中，表示矩阵中一行的结束；也可用于语句后隔开语句。当一条语句后出现分号时，MATLAB 将不给出运行的中间结果。

举例：

在 MATLAB 命令窗口中输入：

```
A = magic(3)
```

MATLAB 将直接给出运行结果，

```
A =
     8     1     6
     3     5     7
     4     9     2
```

当在上面的语句后添加分号后，MATLAB 将暂不给出中间结果，而是等待用户继续输入命令。

10. 注释符

名称：%

注释符。

描述：百分号在 MATLAB 中用于表示注释语句的开始。以百分号开始的语句可以被 HELP 命令显示出来。

11. 感叹号

名称：!

感叹号。

描述：感叹号后的其余部分被作为一个操作系统的命令。在 Macintosh 中不可用。

举例：

在 MATLAB 命令窗口中输入：

```
!dir
```

将显示当前目录，结果如下：

```
 Volume in drive D has no label
 Volume Serial Number is 3B42-1203
 Directory of D:\MATLAB\work
.                 <DIR>          10-15-99   23:04 .
..                <DIR>          10-15-99   23:04 ..
MYFIG      JPG          17,727   10-19-99   21:15 myfig.jpg
DIARY                   21,376   10-28-99   23:41 diary
           2 file(s)          39,103 bytes
           2 dir(s)      195,477,504 bytes free
```

12. 单引号

名称：'

单引号。

语法：该函数有如下两种表达式：

- A'
- A = '...'

描述：在 MATLAB 中，单引号有两种作用，一是转置符，一是括起字符串。当单引号只出现一个时，用作转置符；当单引号成对出现时，用于括起字符串。

举例：

在 MATLAB 命令窗口中输入：

```
A = [1 2 3;4 5 6];
A'
```

结果为：

```
ans =
     1     4
     2     5
     3     6
```

继续输入：

```
A = 'myMATLAB'
```

结果为：

```
A =
myMATLAB
```

13. 数组转置

名称：.'

数组转置。

语法：A.'

描述：A.'命令将对求指定数组的转置数组。

举例：

在 MATLAB 命令窗口中输入：

```
A = [1 2 3;4 5 6];
A.'
```

结果为：

```
ans =
     1     4
     2     5
     3     6
```

14. 赋值号

名称：=

赋值号。

语法：A = B

描述：A = B 命令表示将 B 中的元素存入 A 中。

举例：

在 MATLAB 命令窗口中输入：

```
B = [1 2 3;4 5 6;7 8 9];
A = B;
B
```

结果为：

```
B =
     1     2     3
     4     5     6
     7     8     9
```

2.5 逻辑函数

1. 测试零元素

名称：all

测试是否所有的元素都非零。

语法：该函数有如下两种表达形式：

- B = all(A)
- B = all(A,dim)

描述：B = all(A)命令将按照数组中不同的维测试所有的元素是否都是非零元素，当所有的元素都是非零元素时，则返回逻辑真。

如果 A 是一个向量，当 A 中所有的元素都是非零元素，命令 all(A)将返回逻辑真(1)；当 A 中所有的元素不都是非零元素，命令 all(A)将返回逻辑假(0)。

如果 A 是一个矩阵，命令 all(A)将把该矩阵中的每一列作为一个向量来对待，并返回一个带有若干 0 或 1 的向量。

如果 A 是一个多维数组，命令 all(A)将把沿着第一个非单元素维中的值作为向量对待，并为每个向量返回一个逻辑状态。

B = all(A,dim)命令将测试 A 中用标量"dim"指定的维的所有的元素是否都非零。

举例：

对于一个标量

A=[1 2 3 4 5];

在 MATLAB 命令窗口中输入：

all(A)

结果为

```
ans =
     1
```

对于一个矩阵

```
B =
     1     2     3
     4     5     6
     7     8     0
```

在 MATLAB 命令窗口中输入：

all(B)

结果为

```
ans =
     1     1     0
```

对于一个二维矩阵

```
C =
    0.5200    0.6800         0    3.5000
   12.0000    0.1000   33.0000    2.0000
    6.0000    5.0000    4.0000    3.0000
   32.0000    0.5600    0.7500    9.9000
```

在 MATLAB 命令窗口中输入：

all(C,1)

结果为

```
ans =
     1     1     0     1
```

在 MATLAB 命令窗口中输入：

all(C,2)

结果为

ans =

    0
    1
    1
    1

2. 测试是否存在非零元素

名称：any

测试是否存在非零元素。

语法：该函数有如下两种表达形式：

- B = any(A)
- B = any(A,dim)

描述：B = any(A)命令将沿着 A 的不同维测试是否存在非零元素。若存在任何非零元素，将返回逻辑真。

如果 A 是一个向量，当 A 中存在非零元素时，命令 any(A)将返回逻辑真(1)；当 A 中所有的元素都是 0，命令 any(A)将返回逻辑假(0)。

如果 A 是一个矩阵，命令 any(A)将把该矩阵中的每一列作为一个向量来对待，并返回一个带有若干 0 或 1 的向量。

如果 A 是一个多维数组，命令 any(A)将把沿着第一个非单元素维中的值作为向量对待，并为每个向量返回一个逻辑状态。

B = any(A,dim)测试 A 中用标量“dim”指定的维的所有的元素是否存在非零元素。

举例：

对于一个标量

A=[0 1 2 3 4 5];

在 MATLAB 命令窗口中输入：

any(A)

结果为

ans =

    1

对于一个矩阵

B =

    [1   2   0
     4   5   0
     7   8   0]

在 MATLAB 命令窗口中输入：

any(B)

结果为

ans =

    1   1   0

对于一个三维矩阵

```
C =
    [1    2    0
     3    4    0
     0    0    0
     5    6    0]
```

在 MATLAB 命令窗口中输入：

```
any(C,1)
```

结果为

```
ans =
     1    1    0
```

在 MATLAB 命令窗口中输入：

```
any(C,2)
```

结果为

```
ans =
     1
     1
     0
     1
```

3. 检查文件或变量是否存在

名称：exist

检查文件或变量是否存在。

语法：该函数有如下两种表达形式：

- a = exist('item')
- ident = exist('item','kind')

描述：a = exist('item')命令将把指定的文件或变量的状态返回到“a”中。指定的文件或变量的状态值见表 2-1。

表 2-1　　指定的文件或变量的状态值

状态值	说明
0	指定的文件或变量不存在
1	指定变量存在于工作空间内
2	指定的文件是 M 文件或未知类型的文件
3	指定的文件是 MEX 文件
4	指定的文件是 MDL 文件
5	指定的文件是 MATLAB 内置函数
6	指定的文件是 P 文件
7	指定的是一个目录

ident = exist('item','kind')命令当指定类型的文件或变量被发现时返回逻辑真；当指定类型的文件或变量没有被发现时返回逻辑假。“kind”的值见表 2-2。

表 2-2　　　　参数"kind"的值

"kind"的值	说明
Var	变量
Builtin	内置函数
File	文件
Dir	目录

举例：

在 MATLAB 工作空间内输入：

ident = exist('sin')

结果为

ident = 5

说明"sin"函数是内置函数。

在 MATLAB 工作空间内输入：

ident = exist('A','var')

结果为

ident = 1

说明 MATLAB 工作空间内存在一个名为"A"的变量。

4. 非零元素的索引和值

名称： find

搜索非零元素的索引和值。

语法： 该函数有如下几种表达形式：

- k = find(x)
- [i,j] = find(X)
- [i,j,v] = find(X)

描述： k = find(x)命令将返回数组 x 中非零元素的索引，若数组没有非零元素，则该命令返回一个空矩阵。

[i,j] = find(X)命令将使用向量"i"、"j"分别按行和列返回数组 x 中非零元素的索引，该命令常用于对稀疏矩阵进行操作。

[i,j,v] = find(X)命令将使用向量"i"、"j"分别按行和列返回数组 x 中非零元素的索引，并使用向量"v"返回一个包含所有非零元素的向量。

举例：

在 MATLAB 命令窗口中输入：

X=[0 1 2 1 0];

find(X)

结果为

ans =

　　2　　3　　4

继续输入命令

find(X==0)

结果为

```
ans =
     1     5
```

在 MATLAB 命令窗口中输入：

```
x =
    [1     0     0
     0     1     0
     0     0     0];
[i,j]=find(x)
```

结果为

```
i =
     1
     2
j =
     1
     2
```

在 MATLAB 命令窗口中输入：

```
C =
     1     2     0
     3     4     0
     0     0     0
     5     6     0
```

结果为

```
i =              j =              v =
     1                1                1
     2                1                3
     4                1                5
     1                2                2
     2                2                4
     4                2                6
```

5. 检测状态

名称：is*

检测状态。

语法：该函数有如下几种表达形式：

- k = iscell(C)
- k = iscellstr(S)
- k = ischar(S)
- k = isempty(A)
- k = isequal(A,B,...)
- k = isfield(S,'field')

- TF = isfinite(A)
- k = isglobal(NAME)
- TF = ishandle(H)
- k = ishold
- k = isieee
- TF = isinf(A)
- TF = isletter('str')
- k = islogical(A)
- TF = isnan(A)
- k = isnumeric(A)
- k = isobject(A)
- TF = isprime(A)
- k = isreal(A)
- TF = isspace('str')
- k = issparse(S)
- k = isstruct(S)
- k = isstudent
- k = isunix
- k = isvms

描述：k = iscell(C)当 C 是单元数组时返回逻辑真；当 C 不是单元数组时返回逻辑假。

k = iscellstr(S)当 S 是字符串单元数组时返回逻辑真；当 S 不是字符串单元数组时返回逻辑假。

k = ischar(S)命令当 S 是字符串数组时返回逻辑真；当 S 不是字符串数组时返回逻辑假。

k = isempty(A)命令当 A 是空数组时返回逻辑真；当 A 不是空数组时返回逻辑假。

k = isequal(A,B,...)命令当输入的数组都相等时时返回逻辑真；当输入的数组不相等时返回逻辑假。

k = isfield(S,'field')当“field”是结构数组中的字段时返回逻辑真；否则时返回逻辑假。

TF = isfinite(A)命令将返回一个与 A 有相同大小的数组，该数组由 1 和 0 构成，对于 A 中为有限值的元素，新数组将在与 A 中相同的位置返回 1；对于 A 中为无限值或 NaN 的元素，新数组将在与 A 中相同的位置返回 0。

k = isglobal(NAME)命令当“NAME”是一个全局变量时返回逻辑真；当“NAME”不是一个全局变量时返回逻辑假。

TF = ishandle(H)命令将返回一个与 H 相同大小的数组，对于 H 中是有效图形句柄的元素，新数组将在与 H 中相同的位置返回 1；对于 H 中不是有效图形句柄的元素，新数组将在与 H 中相同的位置返回 0。

k = ishold 命令当“hold”命令处于“on”状态时，返回逻辑真；当“hold”命令处于“off”状态时，返回逻辑假。

k = isieee 命令当用户计算机使用 IEEE 算法时，返回逻辑真；当用户计算机没有使用 IEEE 算法时，返回逻辑假。

TF = isinf(A)命令将返回一个与 A 相同大小的数组，对于 A 中是正负无穷大的元素，新数组将在与 A 中相同的位置返回 1；对于 A 中不是正负无穷大的元素，新数组将在与 A 中相同的位置返回 0。

TF = isletter('str')命令将返回一个与“str”相同大小的数组，对于“str”中是字母表中字母的元素，新数组将在与“str”中相同的位置返回 1；对于“str”中不是字母表中字母的元素，新数组将在与“str”中相同的位置返回 0。

k = islogical(A)命令当 A 是一个逻辑数组时，返回逻辑真；否则返回逻辑假。

TF = isnan(A)命令将返回一个与 A 相同大小的数组，对于 A 中是非数值数的元素，新数组将在与 A 中相同的位置返回 1；对于 A 中不是非数值数的元素，新数组将在与 A 中相同的位置返回 0。

k = isnumeric(A)命令当 A 是一个数值数组时，返回逻辑真；否则返回逻辑假。

k = isobject(A)命令当 A 是一个对象时，返回逻辑真；否则返回逻辑假。

TF = isprime(A)返回一个与 A 相同大小的数组，对于 A 中是质数的元素，新数组将在与 A 中相同的位置返回 1；对于不是质数的元素，新数组将在与 A 中相同的位置返回 0。

k = isreal(A)命令当 A 中的元素都为实数时，返回逻辑真；否则返回逻辑假。

TF = isspace('str')命令将返回一个与“str”相同大小的数组，对于“str”中是 ASCII 空白间隔的元素，新数组将在与“str”中相同的位置返回 1；对于“str”中不是 ASCII 空白间隔的元素，新数组将在与“str”中相同的位置返回 0。ASCII 空白间隔指空格、新行、TAB 键、回车等。

k = issparse(S)命令当 S 的存储类别为稀疏类时返回逻辑真，否则返回逻辑假。

k = isstruct(S)命令当 S 为结构时返回逻辑真，否则返回逻辑假。

k = isstudent 命令当用户的 MATLAB 版本为学生版时，返回逻辑真；当用户的 MATLAB 版本为商业版时，返回逻辑假。

k = isunix 命令当用户的 MATLAB 版本为 UNIX 版时，返回逻辑真；当用户的 MATLAB 版本不是 UNIX 版时，返回逻辑假。

k = isvms 命令当用户的 MATLAB 版本为 VMS 版时，返回逻辑真；当用户的 MATLAB 版本不是 VMS 版时，返回逻辑假。

举例：

在 MATLAB 命令窗口中输入：

```
A = rand(3,4,5);
A(:,:,:) = [];
isempty(A)
```

结果为

```
ans =
     1
```

在 MATLAB 命令窗口中输入：

```
S = 'A1,B2';
isletter(S)
```

结果为

ans =

        1   0   0   1   0

在 MATLAB 命令窗口中输入：

A=

  [0 1

   1 0]

B=

  [1 1

   1 0]

C=

  [0 1

   1 0]

isequal(A,B,C)

结果为

ans =

      0

继续输入

isequal(A,C)

结果为

ans =

      1

6. 检测所给类中的对象

名称：isa

检测所给类中的对象。

语法：K = isa(obj,'class_name')

描述：K = isa(obj,'class_name')命令当指定的对象属于指定的类时，返回逻辑真；当指定的对象不属于指定的类时，返回逻辑假。

参数"class_name"为自定义或预定义的类名。系统预先定义的类名见表 2-3。

表 2-3　　　　系统预先定义的类名

类	说明
cell	多维的单元数组
double	多维的双精度数组
sparse	二维的稀疏数组
char	字符数组
struct	结构
class_name	自定义类

举例：

在 MATLAB 中输入

isa(rand(4),'char')

结果显示为：

```
ans =
      0
```

7. 数字值转换为逻辑值

名称：logical

将数字值转换为逻辑值。

语法：K = logical(A)

描述：K = logical(A)返回能够用于逻辑索引或逻辑测试的数组。当 A 和 B 大小相同且 B 是一个逻辑数组时，命令 A(B)将按照 B 索引的非零元素返回 A 中相同位置的元素的值。

用户也可以使用关系操作符(==,<,>,~, etc.)和类似 any、all、isnan、isinf、isfinite 等函数创建逻辑数组。

举例：

在 MATLAB 命令窗口中输入：

```
A=rand(4)
A =
    0.9355    0.0579    0.1389    0.2722
    0.9169    0.3529    0.2028    0.1988
    0.4103    0.8132    0.1987    0.0153
    0.8936    0.0099    0.6038    0.7468
B = logical(eye(4))
B =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
A(B)
```

结果为

```
ans =
    0.9355
    0.3529
    0.1987
    0.7468
```

8. 如果 M 文件不能被清除时为真

名称：mislocked

如果 M 文件不能被清除时为真。

语法：该函数有如下两种表达形式：

- mislocked
- mislocked(fun)

描述：mislocked 命令当正在运行的 M 文件是被锁定的状态时值为 1；否则其值为 0。mislocked(fun)命令当指定的 M 文件在内存中是被锁定的状态时，其值为 1；否则为 0。

第三章　语言结构和调试命令

3.1　程序设计

1. 执行内置函数

名称：builtin

从过载方法中执行内置函数。

语法：该函数有如下两种表达形式：

- builtin(function,x1,...,xn)
- [y1,..,yn] = builtin(function,x1,...,xn)

描述：builtin 命令常使用那些过载内置函数的方法执行初始的内置函数。若“function”是一个包含内置函数名的字符串，则

builtin(function,x1,...,xn)命令根据所给的参数求函数值。

[y1,..,yn] = builtin(function,x1,...,xn)命令将返回多个输出参数。

2. 执行包含表达式的字符串

名称：eval

执行一个包含表达式的字符串。

语法：该函数有如下几种表达形式：

- eval(expression)
- [a1,a2,a3,...] = eval(expression)
- eval(expression,catch_expr)

描述：eval(expression)命令将执行指定的表达式。用户可以使用方括号连接子串和变量来构造新的表达式 expression = [string1,int2str(var),string2,...]。

[a1,a2,a3,...] =eval(expression)执行指定的表达式并将结果返回到指定的输出变量中。

eval(expression,catch_expr)执行指定的表达式，当检测到错误时，将执行“catch_expr”。

举例：

在 MATLAB 命令窗口中执行一个表达式。

```
A='2*3.14*5';
B=eval(A)
```

结果为

```
B = 31.4000
```

3. 表达式的值

名称：evalc

计算并捕获表达式的值。

语法：该函数有如下两种表达形式：

- T = evalc(S)
- T = evalc(s1,s2)
- [T,X,Y,Z,...] = evalc(S)

描述：T = evalc(S)命令将执行指定的表达式，捕捉正常被写入命令窗口的内容并返回到字符数组T中。

T = evalc(s1,s2)命令与 eval(s1,s2)命令作用基本相同，此外将捕捉正常被写入命令窗口的内容并返回到字符数组 T 中。

[T,X,Y,Z,...] = evalc(S)命令与[X,Y,Z,...] = eval(S)命令作用基本相同，此外将捕捉正常被写入命令窗口的内容并返回到字符数组 T 中。

举例：

在 MATLAB 命令窗口中输入：

T = evalc('2*3.14*5')

结果为：T = 31.4000

4. 执行包含 MATLAB 表达式的字符串

名称：evalin

在工作空间内执行包含 MATLAB 表达式的字符串。

语法：该函数有如下几种表达形式：

- evalin(ws,expression)
- [a1,a2,a3,...] = evalin(ws,expression)
- evalin(ws,expression,catch_expr)

描述：evalin(ws,expression)命令将执行工作空间内指定的表达式。"ws" 的值见表 3-1。

表 3-1 参数 "ws" 的值

"ws"的值	说明
base	基本工作空间
caller	调用程序工作空间

[a1,a2,a3,...] = evalin(ws,expression)命令执行指定工作空间内的表达式并将结果返回到指定的输出变量中。

evalin(ws,expression,catch_expr)命令将执行指定工作空间内的表达式，当检测到一个错误时，将执行 "catch_expr"。

MATLAB 基本工作空间是指从 MATLAB 命令行中能够看到的工作空间；而 MATLAB 调用程序工作空间是指调用 M 文件函数的工作空间。

举例：

下面的例子将在基本工作空间内找到 A 的值并使用变量 "w" 捕捉该值。

A='2*3.14*5';

w = evalin('base','A')

结果为 w =2*3.14*5

5. 函数值

名称：feval

求函数值。

语法：[y1,y2, ...] = feval(function,x1,...,xn)

描述：[y1,y2, ...] = feval(function,x1,...,xn)命令将使用所给的参数求指定函数的值。

举例：

在 MATLAB 命令窗口中输入：

```
A = [0 -1 0;1 0 -1;-1 -1 -1];
x = feval('abs',A)
```

结果为：

```
x =
     0     1     0
     1     0     1
     1     1     1
```

6. M 文件函数

名称：function

M 文件函数。

描述：用户可以利用已有的函数在 MATLAB 词汇表中添加新的函数。在一个被称为 M 文件的文本文件中，用户可以使用已有的函数和命令组成新的函数。

M 文件可以是脚本或函数：脚本是那些包含 MATLAB 语句的较为简单的文件；而函数可以使用本身的局部变量并接受输入参数。

M 文件的名称必须以字母开头，并带有一个“.m”的扩展名。

M 文件的最顶部的行包含了该文件的语法定义。函数名将在 M 文件的第一行中进行定义，并应该与定义该函数的 M 文件名相同。

举例：

例如，在 MATLAB 命令窗口中输入 type var，结果将显示“var.m”文件的内容。其中的第一行为 function y = var(x,w)。

7. 全局变量

名称：global

定义一个全局变量。

语法：global X Y Z

描述：global X Y Z 命令将定义变量“X”、“Y”和“Z”为全局变量。

通常情况下，每一个 MATLAB 函数都有自己的局部变量，这些局部变量的作用范围在该函数体内。当一个变量被声明为一个全局变量时，则 MATLAB 工作空间内的函数将共同使用该变量。当用户没有使用命令声明全局变量时，系统将全局变量用一个空矩阵进行初始化。当用户声明一个全局变量，而当前的工作空间内已经存在一个与要声明的全局变量同名的变量时，MATLAB 将发出一个警告并使用该变量的值作为全局变量的值。用户可以使用 clear global variablename 命令来从全局工作空间内清除一个指定的全局变量。使用该命令将不会影响该变量的值。在一个回调语句中使用全局变量后，应该使用 clear global variablename 命令从当前的工作空间内清除该全局变量，这样做的目的是避免引用该全局变量后才声明全局变量的情况出现。

举例：

在 MATLAB 命令窗口中输入：

A=1;

global A

结果为

Warning: The value of local variables may have been changed to match the globals.Future versions of MATLAB will require that you declarea variable to be global before you use that variable.。此时，将把原先的变量 A 声明为全局变量。

8. 输入参数的数字

名称：nargchk

检查输入参数的数字。

语法：msg = nargchk(low,high,number)

描述：nargchk 命令常用于在 M 文件中检查被传递的参数数目是否正确。

msg = nargchk(low,high,number)命令当检查到被传递的参数数目“number”小于“low”或大于“high”时将返回一个错误信息；当检查到被传递的参数数目“number”大于“low”且小于“high”时将返回一个空矩阵。

9. 常量

名称：persistent

定义常量。

语法：persistent X Y Z

描述：persistent X Y Z 把“X”、“Y”和“Z”定义为常量。

当 M 文件从内存中被清除或 M 文件被改变时，M 文件中的常量将被清除。若需要让一个 M 文件一直保存在内存中，可以使用 mlock 文件。按照惯例，常量名通常使用大写字母。

10. 脚本 M 文件

名称：script

作为脚本的 M 文件。

描述：一个脚本文件是一个包含 MATLAB 语句序列的外部文件。脚本文件有“.m”的扩展名并被称为 M 文件。脚本文件是最简单的 M 文件，并用于自动执行 MATLAB 命令块。脚本能够操作工作空间内存在的数据，也可以创建新的数据。尽管脚本不能返回输出参数，但其创建的变量能够保存在工作空间内，以便用户能够使用这些变量进行更多的计算。此外，脚本能够使用类似“plot”的命令来产生图形输出。脚本能够包含连续的 MATLAB 语句，在脚本中，这些语句不需要声明或“begin/end”分界符。同任何 M 文件一样，脚本能够包含注释。任何以百分号“%”开始的行都是注释文本。

3.2 流程控制

1. 停止执行“for”循环或“while”循环

名称：break

停止执行“for”循环或“while”循环。

语法：break

描述：break 命令停止执行“for”循环或“while”循环。在嵌套循环中，break 命令将从最里面的循环退出。

2. 情况语句

名称：case

情况语句。

语法：case

描述：case 语句是 switch 语句语法中的一部分。该语句允许按照条件执行。

举例：

switch-case 语句的一般格式为：

```
switch switch_expr
      case case_expr
         statement,...,statement
      case {case_expr1,case_expr2,case_expr3,...}
         statement,...,statement
...
      otherwise
         statement,...,statement
end
```

3. 捕捉块

名称：catch

开始执行捕捉块。

语法：catch

描述：catch 开始执行捕捉程序。try-catch 语句的常用格式如下：

```
try statement,
....
statement,
catch
statement,
....
statement
end
```

通常，只有 try 和 catch 间的语句才执行。然而，当执行这些语句发生一个错误时，错误将被捕捉到 lasterr 中，并且 catch 和 end 间的语句将被执行。

4. 有条件地执行语句(else)

名称：else

有条件地执行语句。

语法：if expression

```
        statements
    else
        statements
    end
```

描述：else 命令常用于描述一个交替的语句块。

若第一个表达式为假且第二个表达式为真，将执行 else 命令后的语句块。

表达式通常是下面语句的结果：expression rop expression，其中，rop 见表 3-2。

表 3-2 MATLAB 中的关系符

rop	说明
==	等于
>	大于
<	小于
>=	大于等于
<=	小于等于
~=	不等于

5. 有条件地执行语句(elseif)

名称：elseif

有条件地执行语句。

语法：

```
if expression
        statements
elseif expression
        statements
end
```

描述：elseif 命令将根据条件执行语句。

若第一个表达式为假且第二个表达式为真，将执行 elseif 命令后的语句块。

表达式通常是下面语句的结果：expression rop expression

请读者注意，else if 命令与 elseif 命令的作用是有所区别的。使用 else if 命令将引入一个新的、嵌套的 if 语句，并且必须和 end 命令匹配使用。

举例：

下面是分别使用 else if 命令和 elseif 命令的例子。

使用 else if 命令的例子：

```
if A
    x=1
else
    if B
        x=2
    else
        if C
            x=3
        else
```

```
            x=4
        end
    end
end
```

使用 elseif 命令的例子：

```
if A
    x=1
elseif B
    x=2
elseif C
    x=3
else
    x=4
end
```

这两个程序段的执行结果相同。

6. 终止 for、while、switch、try 和 if 语句

名称：end

终止 for、while、switch、try 和 if 语句或指出最后一个索引。

语法：while expression

```
    statements
end
B = A(index:end,index)
```

描述：end 命令用于终止 for、while、switch、try 和 if 语句。没有 end 命令，for、while、switch、try 和 if 语句将等待更多的输入。end 命令也用作一个索引表达式的最后一个索引。

7. 显示错误信息

名称：error

显示错误信息。

语法：error('error_message')

描述：error('error_message')命令将显示一个错误信息并返回到键盘控制。错误信息包含在"error_message"中。当 error('error_message')命令中的"error_message"是一个空字符串时，该命令将不产生任何作用。

8. 重复执行语句

名称：for

按照指定的次数重复执行语句。

语法：for variable = expression

```
    statements
end
```

描述：for 命令的常用格式为：

for variable = expression

```
    statement
    ...
    statement
end
```

举例:

下面的例子是一个使用 for 循环创建希尔伯特矩阵的例子。

```
a = zeros(5,5)
for i = 1:5
    for j = 1:5
        a(i,j) = 1/(i+j -1);
    end
end
```

9. 有条件地执行语句

名称：if

有条件地执行语句。

语法：有如下几种表达形式：

```
if expression
    statements
end
if expression1
    statements
elseif expression2
    statements
else
    statements
end
```

描述：if 命令将有条件地执行语句。if 命令最简单的格式如下：

```
if expression
    statements
end
```

10. switch 语句的一部分

名称：otherwise

switch 语句的一部分。

描述：otherwise 命令是 switch 语句的一部分，该语句允许按照条件执行语句。仅当 otherwise 命令前的所有 case 表达式都与 switch 表达式不匹配时，才执行该命令后的语句。

举例:

使用 otherwise 命令的通用格式为：

```
switch sw_expr
    case case_expr
```

```
            statement
            statement
        case {case_expr1,case_expr2,case_expr3}
            statement
            statement
        otherwise
            statement
            statement
end
```

11. 返回到调用函数

名称：return

返回到调用函数。

语法：return

描述：return 命令将正常返回到调用函数或键盘输入。该命令也可用于终止键盘模式。

12. 开关语句

名称：switch

基于一个条件表达式的开关语句。

语法：switch switch_expr

```
  case case_expr
     statements
  case {case_expr1,case_expr2,case_expr3,...}
     statements
...
  otherwise
     statements
end
```

描述：switch 命令的语法是有条件执行代码语句的方式之一。switch 语句将首先计算表达式 switch_expr 的值，并使用该值按照次序和每个 case_expr 表达式进行比较，最先与 switch_expr 表达式匹配的 case_expr 表达式后的语句将被执行。

如果所有的 case_expr 表达式都不能和 switch_expr 表达式相匹配，otherwise 语句后的代码将被执行。表达式 switch_expr 可以是一个标量或一个字符串。

13. 执行 try 程序块

名称：try

开始执行 try 程序块。

描述：该命令的通用格式为：

```
try
statement
......
statement,
```

```
catch
statement
......
statement
end
```

通常，只有 try 和 catch 间的语句才执行。然而，当执行这些语句发生一个错误时，错误将被捕捉到 lasterr 中，并且 catch 和 end 间的语句将被执行。如果 catch 命令后的语句中再次产生错误，将停止执行语句。

14. 警告信息

名称：warning

显示警告信息。

语法：有如下几种表达形式：

- warning('message')
- warning on
- warning off
- warning backtrace
- warning debug
- warning once
- warning always
- [s,f] = warning

描述：warning('message')命令当执行 disp 函数时显示“message”文本信息。

warning on 命令将抑制该命令后的所有警告信息。

warning off 命令将允许显示警告信息。

warning backtrace 除了与 warning 命令有相同的作用外，并将引发警告的文件及行号显示出来。

warning debug 命令当遇到一个警告时，启动调试程序。

warning once 命令在每个会话中仅显示一次图形句柄向后兼容警告。

warning always 命令在每个会话中将一直显示图形句柄向后兼容警告。

[s,f] = warning 命令将使用字符串 s 返回当前的警告状态，并使用字符串 f 返回当前的警告发生次数。

15. 不确定次数地重复执行语句

名称：while

不确定次数地重复执行语句。

语法：while expression

```
    statements
end
```

描述：while 命令将不确定次数的重复执行语句。

举例：

在 MATLAB 命令窗口中输入：

```
i=1;
while i<100
    i=i+1
end
```

回车后，这段代码将一直执行语句 i=i+1，直到 i = 100 时为止。

3.3 交互输入

1. 要求用户输入

名称：input

要求用户输入。

语法：有如下两种表达形式：

- user_entry = input('prompt')
- user_entry = input('prompt','s')

描述：user_entry = input('prompt')命令将在屏幕上显示一个提示符，等待用户从键盘的输入，并将输入的值返回到变量“user_entry”中。

user_entry = input('prompt','s')命令将在屏幕上显示一个提示符，等待用户从键盘的输入，并将输入的字符串作为一个文本变量返回到“user_entry”中。如果用户在提示符后直接按回车键，input 命令将返回一个空矩阵到变量“user_entry”中。

2. 在 M 文件中调用键盘

名称：keyboard

在一个 M 文件中调用键盘。

语法：keyboard

描述：当 keyboard 命令出现在一个 M 文件中时，该文件将停止执行而将控制交给键盘，并产生一个以 K 开头的提示符。用户可以在提示符后检查或改变变量。在提示符后输入的任何 MATLAB 命令都是有效的。该命令模式对于用户调试自己的 M 文件是非常有用的。

若需要终止键盘输入模式，在提示符后直接键入 return，并直接按回车键。

3. 选择菜单

名称：menu

为用户输入产生一个选择菜单。

语法：k = menu('mtitle','opt1','opt2',...,'optn')

描述：k = menu('mtitle','opt1','opt2',...,'optn')命令将弹出一个菜单，其标题是“mtitle”中的字符串；选项为“opt1”、“opt2”到“optn”中的字符串。要从其他的用户对象中调用 menu 命令，可以设置该用户对象的“Interruptible”属性为“YES”。

举例：

在 MATLAB 命令窗口中输入：

```
k = menu('你好！这是一个选择菜单，用户可以按下不同的按钮来显示不同的图案。',
    '按钮 1','按钮 2','按钮 3','按钮 4')
```

结果如图 3-1 所示。

当用户单击“按钮 1”时，MATLAB 命令窗口中将显示：

k =　1

并关闭该菜单。

同样当用户分别单击“按钮 2”、“按钮 3”和“按钮 4”时，在 MATLAB 命令窗口中将分别返回 k=2、3、4；当用户直接关闭该菜单时，MATLAB 命令窗口中将返回 k=0。

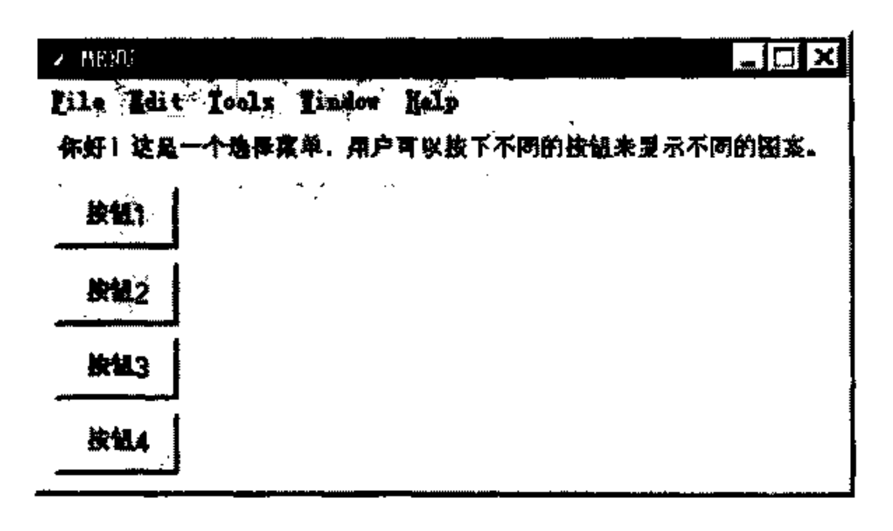


图 3-1　一个选择菜单

4. 临时暂停执行

名称：pause

临时暂停执行。

语法：有如下几种表达形式：

- pause
- pause(n)
- pause on
- pause off

描述：pause 命令将暂停执行 M 文件，当用户按任意键时将继续执行暂停的 M 文件。

pause(n)命令将用“n”来指定暂停的秒数，其中，“n”必须为实数。

pause on 命令将允许该命令后的暂停命令产生作用。

pause off 命令将不允许该命令后的暂停命令产生作用。

3.4 面向对象编程

1. 创建对象

名称：class

创建一个对象或返回一个对象的类。

语法：有如下几种表达形式：

- str = class(object)

- obj = class(s,'class_name')
- obj = class(s,'class_name',parent1,parent2...)

描述：str = class(object)使用字符串指定对象的类。MATLAB 中对象的类见表 3-3。

表 3-3　　　　MATLAB 中对象的类

对象的类	说明
Cell	多维单元数组
Double	多维双精度数组
Sparse	二维稀疏数组
Char	字符数组
Struct	结构
'class_name'	自定义类

obj = class(s,'class_name')把结构"s"作为一个模板并根据指定的类名来创建一个对象。

obj = class(s,'class_name',parent1,parent2...)命令将把结构"s"作为一个模板，根据指定的类名来创建一个对象，并确保新创建的对象继承了指定父类的方法。

2. 双精度

名称：double

转换为双精度。

语法：double(X)

描述：double(X)返回指定数组 X 的双精度值。若 X 已经是一个双精度数组，该命令将不产生作用。double 命令常常在循环结构中那些已经不是双精度的表达式中被调用。

举例：

在 MATLAB 命令窗口中输入：

```
A = [1.2 3;5 -5;0 3.14];
B = double(A)
```

结果为：

```
B =
    1.2000    3.0000
    5.0000   -5.0000
         0    3.1400
```

3. 亚类关系

名称：inferiorto

亚类关系

语法：inferiorto('class1','class2',...)

描述：inferiorto 函数将建立一个层次，该层次将决定 MATLAB 调用类方法的次序。

如果 class1，class2 等类的一个或多个对象调用一个函数时，在类结构方法内被调用的 inferiorto('class1','class2',...)命令将指出该类的方法不应被调用。

如果一个函数被没有指定层次关系的两个对象调用，这两个对象将被认为优先级相同，并且处于左边的对象的方法将被调用。

举例：

假设 A 为类"class_a"中的对象，B 为类"class_b"中的对象，C 为类"class_c"中的

对象，同时假设结构“class_c.m”包含语句：

inferiorto('class_a')

则命令 x = fun(a,c)或 x = fun(c,a)将调用类“class_a”中的函数。

4. 内联对象

名称：inline

构建一个内联的对象。

语法：有如下几种表达形式：

- g = inline(expr)
- g = inline(expr,arg1,arg2, ...)
- g = inline(expr,n)

描述：g = inline(expr)命令将根据指定的表达式构建一个内联函数。

g = inline(expr,arg1,arg2, ...)命令将根据指定的表达式构建一个内联函数，其输入参数使用 arg1、arg2 等等来指定。

g = inline(expr,n)命令将根据指定的表达式构建一个内联函数，其输入参数是 x，P1，P2 等，其中 n 为一个标量。

三个与 inline 命令相关的命令允许用户检查一个内联函数对象，并决定这个内联函数对象如何被创建。

char(fun)命令把内联函数转化到一个字符数组。

argnames(fun)命令把内联对象的输入参数名作为一个字符串单元数组返回。

formula(fun)命令返回内联对象的公式。

举例：

下面将创建一个简单的内联函数，该函数可以对一个数字进行立方。在 MATLAB 命令窗口中输入 g = inline('t^3')，结果显示为：

```
g =   Inline function:
      g(t) = t^3
```

用户也可以使用 char 命令将结果转换为一个字符串。在 MATLAB 命令窗口中输入：

```
char(g)
```

结果显示

```
ans =   t^3
```

下面的例子将创建一个内联函数来描绘公式 g(x) = 2*cos(2*x)，在 MATLAB 命令窗口中输入：

```
g = inline('2*cos(2*x)')
```

结果显示

```
g =
      Inline function:
      g(x) = 2*cos(2*x)
```

在 MATLAB 命令窗口中继续输入：

```
argnames(g)
```

结果为

ans =

'x'

在 MATLAB 命令窗口中继续输入：

formula(g)

结果为

ans =

2*cos(2*x)

5. 转换到带符号整数

名称：int8，int16，int32

转换到带符号整数。

语法：有如下几种表达形式：

- i = int8(x)
- i = int16(x)
- i = int32(x)

描述：i = int8(x)命令将向量 x 转换为带符号的 8 位整数。

i = int16(x)命令将向量 x 转换为带符号的 16 位整数。

i = int32(x)命令将向量 x 转换为带符号的 32 位整数。int 命令的操作结果见表 3-4。

表 3-4　　int 命令的操作结果

操作命令	输出范围	输出类型	每个元素所占字节数	输出类
int8	-128 到 127	带符号 8 位整数	1	int8
int16	-32768 到 32767	带符号 16 位整数	2	int16
int32	-2147483648 到 2147483647	带符号 32 位整数	4	int32

举例：

在 MATLAB 命令窗口中输入：

x = [-1.2358 0 3.1415 168];

i = int8(x)

结果为：

i =

-1　　0　　3　　127

6. 检测对象

名称：isa

检测对象是否为所给的类的对象。

语法：K = isa(obj,'class_name')

描述：K = isa(obj,'class_name')命令当指定的对象是所给的类的实例时返回逻辑真；否则返回逻辑假。

举例：

在 MATLAB 命令窗口中输入 K = isa(rand(4),'double')，

结果显示

K = 1。

7. load 函数用户定义扩展

名称：loadobj

load 函数的用户定义扩展。

语法：b = loadobj(a)

描述：b = loadobj(a)命令为用户对象扩展 load 函数。输入参数 a 是一个 MAT 文件载入的对象，输出参数 b 是 load 函数将要载入到工作空间内的对象。

8. save 函数的用户定义扩展

名称：saveobj

save 函数的用户定义扩展。

语法：b = saveobj(a)

描述：b = saveobj(a)对用户对象扩展 save 函数。输入参数 a 是 MATLAB 工作空间内的对象，b 是 save 函数保存到 MAT 文件中的对象。saveobj 仅能够用于用户对象的过载。

9. 单精度

名称：single

转换为单精度。

语法：Y = single(X)

描述：Y=single(X)转换向量 X 为单精度。X 可为任何数字的对象。如果 X 已经是单精度，该命令将没有作用。单精度数比双精度数需要较少的存储空间，但精度更低且范围更小。

举例：

在 MATLAB 命令窗口中输入：

```
x = [-378968 -1.23456789;0 0.25;67895623 3.1415];
y = single(x)
```

结果为：

```
y =
   1.0e+007 *
    -0.0379    -0.0000
          0     0.0000
     6.7896     0.0000
```

10. 超类关系

名称：superiorto

超类关系。

语法：superiorto('class1','class2',...)

描述：superiorto 函数将建立一个层次，该层次将决定 MATLAB 调用类方法的次序。如果 class1，class2 等类的一个或多个对象调用一个函数时，在类结构方法内被调用的 superiorto('class1','class2',...)命令将指出该类的方法应被调用。

如果一个函数被没有指定层次关系的两个对象调用，这两个对象将被认为优先级相同，并且处于左边的对象的方法将被调用。

11. 无符号整数

名称：uint8，uint16，uint32

转换到无符号整数。

语法：有如下几种表达形式：

- i = int8(x)
- i = int16(x)
- i = int32(x)

描述：i = uint8(x)命令将把向量 x 转换为无符号 8 位整数。

i = uint16(x)命令将把向量 x 转换为无符号 16 位整数。

i = uint32(x)命令将把向量 x 转换为无符号 32 位整数。uint 命令的操作结果见表 3-5。

表 3-5　　uint 命令的操作结果

操作命令	输出范围	输出类型	每个元素所占字节数	输出类
uint8	0 到 255	无符号 8 位整数	1	uint8
uint16	0 到 65535	无符号 16 位整数	2	uint16
uint32	0 到 4294967295	无符号 32 位整数	4	uint32

当 x 中的一个值超出所在类的范围时，其值将被映射到该类范围的一个端点。如果 x 已经是一个带符号的整数，这些命令将不产生作用。

举例：

在 MATLAB 命令窗口中输入：

```
x = [-378968 -1.23456789;0 0.25;67895623 3.1415];
i = uint8(x)
```

结果为：

```
i =    0      0
       0      0
     255      3
```

继续输入 i = uint16(x)，结果为：

```
i =          0          0
             0          0
         65535          3
```

继续输入 i = uint32(x)，结果为：

```
i =          0          0
             0          0
      67895623          3
```

3.5 程序调试

1. 清除断点

名称：dbclear

清除断点。

语法：有如下几种表达形式：

- dbclear all
- dbclear all in mfile
- dbclear in mfile
- dbclear in mfile at lineno
- dbclear in mfile at subfun
- dbclear if error
- dbclear if warning
- dbclear if naninf
- dbclear if infnan

描述：dbclear all 命令将从所有 MATLAB 文件中删除所有断点。

dbclear all in mfile 命令将从所有".m"文件中删除所有断点。

dbclear in mfile 命令将删除设置在".m"文件中第一个可执行行的断点。

dbclear in mfile at lineno 命令将删除设置在".m"文件中指定行号的断点。

dbclear in mfile at subfun 命令将删除设置在".m"文件中子函数的断点。

dbclear if error 命令将删除使用"dbstop"命令产生错误时引发的断点。

dbclear if warning 命令将删除使用"dbstop"命令产生警告时引发的断点。

dbclear if naninf 命令将删除使用"dbstop"命令产生 naninf 时引发的断点。

dbclear if infnan 命令将删除使用"dbstop"命令产生 infnan 时引发的断点。

2. 重新开始执行

名称：dbcont

重新开始执行。

语法：dbcont

描述：dbcont 从一个 MATLAB 文件的断点处继续执行该文件。当遇到文件中的另一个断点、发生一个错误或 MATLAB 返回到基本工作空间提示符时，执行将再次停止。

3. 工作空间环境

名称：dbdown

改变本地工作空间环境。

语法：dbdown

描述：dbdown 命令当系统遇到一个断点时，将把当前的工作空间改变为调用 MATLAB 文件的工作空间。只有用户已经调用至少一次 dbup 命令后，才可以使用 dbdown 命令。这两个命令的作用是相反的。

4. 调试 MEX 文件

名称：dbmex

允许调试 MEX 文件。

语法：有如下几种表达形式：

- dbmex on
- dbmex off
- dbmex stop
- dbmex print

描述：dbmex on 命令将允许在 UNIX 平台上调试 MEX 文件。

dbmex off 命令将禁止调试 MEX 文件。

dbmex stop 命令将返回到调试程序提示符。

dbmex print 命令将显示 MEX 调试信息。

在 UNIX 平台上调试 MEX 文件时，必须在一个调试程序中输入命令 matlab –Ddebugger，其中，“debugger”字符是调试程序的名称。

5. 退出调试模式

名称：dbquit

退出调试模式。

语法：dbquit

描述：dbquit 命令将立即终止调试程序并将控制返回给基本工作空间提示符。使用 dbquit 命令退出调试程序，被操作的 MATLAB 文件可能会不完整，且没有返回结果。

6. 堆栈

名称：dbstack

显示函数调用堆栈。

语法：有如下两种表达形式：

- dbstack
- [ST,I] = dbstack

描述：dbstack 显示导致当前断点产生的调用函数的名称及行数，并按照它们执行的次序将之列出。

[ST,I] = dbstack 命令将使用一个带有表 3-6 列出字段的结构 ST 来返回堆栈追踪信息，并使用 I 来返回当前的工作空间索引。

表 3-6　　结构 ST 中的字段

字段	说明
name	函数名
Line	函数行数

7. 断点

名称：dbstatus

列出所有断点。

语法：有如下几种表达形式：

- dbstatus
- dbstatus function
- s = dbstatus(...)

描述：dbstatus 命令将列出所有断点，其中包括错误、警告及 naninf。

dbstatus function 命令将显示一个关于指定 MATLAB 文件中的断点的列表。

s = dbstatus(...)命令将使用一个带有表 3-7 列出字段的结构 s 来返回断点信息。

用户可以使用如下三个命令

dbstatus class/function

dbstatus private/function

dbstatus class/private/function

来分别决定方法、私有函数和私有方法的断点状态。

表 3-7　　结构 s 中的字段

字段	说明
name	函数名
line	函数行数
cond	条件字符串(error，warning，或 naninf)

8．从断点处执行语句

名称：dbstep

从一个断点处执行一行或多行语句。

语法：有如下几种表达形式：

- dbstep
- dbstep nlines
- dbstep in

描述：dbstep 命令将执行当前 M 文件断点处的下一可执行行。

dbstep nlines 命令将执行当前 M 文件的指定执行行。

dbstep in 跳到当前 M 文件断点处的下一可执行行。若该行包含对另一个 M 文件的调用，将从被调用的 M 文件的第一个可执行行继续执行；如果该行没有调用其他 M 文件，该命令的作用与 dbstep 相同。dbstep 允许用户从当前断点按照文件执行顺序来调试一个 M 文件。

9．设置断点

名称：dbstop

在一个 MATLAB 函数中设置断点。

语法：有如下几种表达形式：

- dbstop in mfile
- dbstop in mfile at lineno
- dbstop in mfile at subfun
- dbstop if error
- dbstop if warning
- dbstop if naninf
- dbstop if infnan

描述：dbstop in mfile 命令将暂时停止 M 文件的执行，在最先遇到的可执行行，把 MATLAB 设置为调试模式。

dbstop in mfile at lineno 命令将暂时停止 M 文件的执行，在行号为 lineno 的可执行行前，把 MATLAB 设置为调试模式。

dbstop in mfile at subfun 命令将暂时停止 M 文件的执行，在执行用 subfun 指定的子函数前，把 MATLAB 设置为调试模式。

dbstop if error 命令当用户运行的 M 文件产生任何运行错误时，停止执行 M 文件，将 MATLAB 设置为调试模式并停留在产生错误的那一行。

dbstop if warning 命令当用户运行的 M 文件产生任何运行警告时，停止执行 M 文件，

将 MATLAB 设置为调试模式并停留在产生警告的那一行。

dbstop if naninf 命令当用户运行的 M 文件遇到 NaN 时，停止执行 M 文件，将 MATLAB 设置为调试模式并停留在遇到 NaN 的那一行。

dbstop if infnan 命令当用户运行的 M 文件遇到 Inf 时，停止执行 M 文件，将 MATLAB 设置为调试模式并停留在遇到 Inf 的那一行。

10. M 文件内容

名称：dbtype

列出带行号的 M 文件内容。

语法：有如下两种表达形式：

- dbtype function
- dbtype function start:end

描述：dbtype function 命令将指定的 MATLAB 函数的行号及内容。

dbtype function start:end 命令将显示指定 MATLAB 函数中用行号规定范围的部分。

11. 改变本地工作空间环境

名称：dbup

改变本地工作空间环境。

语法：dbup

描述：dbup 命令当系统遇到一个断点时，将把当前的工作空间改变为调用 MATLAB 文件的工作空间。

该命令允许用户使用其他 MATLAB 命令检查正在调用的 M 文件。通过这种方式，用户可以传递参数到被调用的函数中。

第四章　矩阵和矩阵操作基础

4.1　矩阵和数组基础

1. 分块对角矩阵

名称：blkdiag

根据输入构造一个分块对角矩阵。

语法：out = blkdiag(a,b,c,d,...)

描述：out = blkdiag(a,b,c,d,...)命令将根据输入的“a”、“b”、“c”、“d”等参数来构造一个分块对角矩阵。作为输入参数的矩阵不必是方阵，也不必具有相等的大小。

举例：

在 MATLAB 命令窗口中输入如下命令：

```
a=1;
b=[2,2;3,3];
c=[4,4;5,5;6,6];
d=8;
out = blkdiag(a,b,c,d)
```

结果为

```
out =
     1     0     0     0     0     0
     0     2     2     0     0     0
     0     3     3     0     0     0
     0     0     0     4     4     0
     0     0     0     5     5     0
     0     0     0     6     6     0
     0     0     0     0     0     8
```

2. 单位矩阵

名称：eye

单位矩阵。

语法：有如下几种表达形式：

- Y = eye(n)
- Y = eye(m,n)
- Y = eye(size(A))

描述：Y = eye(n)命令将返回一个 n×n 的单位矩阵。

Y = eye(m,n)命令将返回一个 m×n 的单位矩阵。

Y = eye(size(A))命令将返回一个同指定矩阵 A 大小相同的单位矩阵。

该命令不能用于定义多维数组。当试图使用该命令定义一个多维数组时，系统将返回一个错误。

举例：

在 MATLAB 命令窗口中输入：

```
y = eye(3)
```

结果显示为

```
y =
      1     0     0
      0     1     0
      0     0     1
```

在 MATLAB 命令窗口中输入：

```
y = eye(3,2)
```

结果显示为

```
y =
      1     0
      0     1
      0     0
```

在 MATLAB 命令窗口中输入：

```
A =
    [1     2     3
      2     3     4
      3     4     5
      4     5     6]
y = eye(size(A))
```

结果显示为

```
y =
      1     0     0
      0     1     0
      0     0     1
      0     0     0
```

3. 线性间隔向量

名称： linspace

产生线性间隔向量。

语法： 有如下两种表达形式：

- y = linspace(a,b)
- y = linspace(a,b,n)

描述： y = linspace(a,b)命令将产生一个行向量 y，该向量是由把 a 和 b 间的数平分为 100

份而得到的。

y = linspace(a,b,n)产生一个行向量 y，该向量是由把 a 和 b 间的数平分为 n 份而得到的。

举例：

在 MATLAB 命令窗口中输入 y = linspace(-1024,2048,10)，

结果为：

```
y =
  1.0e+003 *
  Columns 1 through 7
   -1.0240   -0.6827   -0.3413         0    0.3413    0.6827    1.0240
  Columns 8 through 10
    1.3653    1.7067    2.0480
```

4. 对数间隔向量

名称：logspace

产生对数间隔向量。

语法：有如下几种表达形式：

- y = logspace(a,b)
- y = logspace(a,b,n)
- y = logspace(a,pi)

描述：y = logspace(a,b)命令将产生一个行向量 y，该向量是由 50 个 10^a 和 10^b 间的对数间隔点构成。

y = logspace(a,b,n)产生一个行向量 y，该向量由 n 个 10^a 和 10^b 间的对数间隔点构成。

y = logspace(a,pi)命令将产生一个行向量 y，该向量是由 50 个 10^a 和 pi 间的对数间隔点构成。该命令常用于数字信号处理领域。该命令的所有参数必须为标量。

举例：

在 MATLAB 命令窗口中输入 y = logspace(2,3,16)，

结果为：

```
y =
  1.0e+003 *
  Columns 1 through 7
    0.1000    0.1166    0.1359    0.1585    0.1848    0.2154    0.2512
  Columns 8 through 14
    0.2929    0.3415    0.3981    0.4642    0.5412    0.6310    0.7356
  Columns 15 through 16
    0.8577    1.0000
```

5. 产生一个元素全为 1 的数组

名称：ones

产生一个元素全为 1 的数组。

语法：有如下几种表达形式：

- Y = ones(n)

- Y = ones(m,n)
- Y = ones([m n])
- Y = ones(d1,d2,d3...)
- Y = ones([d1 d2 d3...])
- Y = ones(size(A))

描述：Y = ones(n)命令将产生一个元素全为 1 的 n×n 数组，当 n 不是标量时，该命令的执行结果将返回一个错误。

Y = ones(m,n)命令将产生一个元素全为 1 的 m×n 数组。

Y = ones([m n])命令同样将产生一个元素全为 1 的 m×n 数组。

Y = ones(d1,d2,d3...)命令将产生一个指定维数、元素全为 1 的多维数组。

Y = ones([d1 d2 d3...])命令同样将产生一个指定维数、元素全为 1 的多维数组。

Y = ones(size(A))命令将返回一个同指定数组 A 同维的元素全为 1 的数组。

举例：

在 MATLAB 命令窗口中输入 Y = ones(3)，

结果显示为：

```
Y =
      1     1     1
      1     1     1
      1     1     1
```

在 MATLAB 命令窗口中输入 Y = ones(3,2)，

结果显示为

```
Y =
      1     1
      1     1
      1     1
```

在 MATLAB 命令窗口中输入：

```
A = [1 2 3;4 5 6];
Y = ones(size(A))
```

结果显示为

```
Y =
      1     1     1
      1     1     1
```

6. 随机数和数组

名称：rand

产生均匀分布随机数和数组。

语法：有如下几种表达式：

- Y = rand(n)
- Y = rand(m,n)
- Y = rand([m n])

- Y = rand(m,n,p,...)
- Y = rand([m n p...])
- Y = rand(size(A))
- s = rand('state')

描述：rand 命令将返回一个随机标量。

Y = rand(n)返回一个 n×n 随机矩阵。如果 n 不是一个标量，该命令将产生一个错误。

Y = rand(m,n)命令将返回一个 m×n 随机矩阵。

Y = rand([m n])命令将同样返回一个 m×n 随机矩阵。

Y = rand(m,n,p,...)命令将返回一个指定维数的随机数组。

Y = rand([m n p...])命令将返回一个指定维数的随机数组。

Y = rand(size(A))命令将返回一个与 A 大小相同的随机数组。

s = rand('state')命令将返回一个包含 35 个元素的向量，该向量显示了发生程序的当前状态。

举例：

在 MATLAB 命令窗口中输入 rand，

结果显示为 ans = 0.3420。

在 MATLAB 命令窗口中输入 rand(3)，

结果显示为：

```
ans =
    0.2897    0.7271    0.5681
    0.3412    0.3093    0.3704
    0.5341    0.8385    0.7027
```

在 MATLAB 命令窗口中输入 rand(2,3)，

结果显示为：

```
ans =
    0.5466    0.6946    0.7948
    0.4449    0.6213    0.9568
```

在 MATLAB 命令窗口中输入：

```
A=[1 2;3 4;5 6];
rand(size(A))
```

结果显示为：

```
ans =
    0.5226    0.9797
    0.8801    0.2714
    0.1730    0.2523
```

7. 正态分布随机数和数组

名称：randn

产生正态分布随机数和数组。

语法：有如下几种表达形式：

- randn
- Y = randn(n)
- Y = randn(m,n)
- Y = randn([m n])
- Y = randn(m,n,p,...)
- Y = randn([m n p...])
- Y = randn(size(A))
- s = randn('state')

描述：randn 命令将返回一个随机标量。

Y = randn(n)返回一个 n×n 随机矩阵。如果 n 不是一个标量，该命令将产生一个错误。

Y = randn(m,n)命令将返回一个 m×n 随机矩阵。

Y = randn([m n])命令将同样返回一个 m×n 随机矩阵。

Y = randn(m,n,p,...)命令将返回一个指定维数的随机数组。

Y = randn([m n p...])命令将同样返回一个指定维数的随机数组。

Y = randn(size(A))命令将返回一个与 A 大小相同的随机数组。

s = randn('state')

举例：

在 MATLAB 命令窗口中输入 randn，

结果显示为 ans = 1.4151。

在 MATLAB 命令窗口中输入 randn(2)，

结果显示为：

```
ans =
   -0.8051    0.2193
    0.5287   -0.9219
```

在 MATLAB 命令窗口中输入 randn(2,3)，

结果显示为：

```
ans =
   -2.1707   -1.0106    0.5077
   -0.0592    0.6145    1.6924
```

在 MATLAB 命令窗口中输入：

```
A=[1 2;3 4;5 6];
randn(size(A))
```

结果显示为：

```
ans =
    0.5913   -1.0091
   -0.6436   -0.0195
    0.3803   -0.0482
```

8. 元素全为 0 的数组

名称：zeros

创建一个元素全为 0 的数组。

语法：有如下几种表达形式：

- B = zeros(n)
- B = zeros(m,n)
- B = zeros([m n])
- B = zeros(d1,d2,d3...)
- B = zeros([d1 d2 d3...])
- B = zeros(size(A))

描述：B = zeros(n)产生一个 n×n 随机矩阵。若 n 不是标量，该命令将产生一个错误。

B = zeros(m,n)命令将返回一个 m×n 随机矩阵。

B = zeros([m n])命令将同样返回一个 m×n 随机矩阵。

B = zeros(d1,d2,d3...)命令将返回一个指定维数的随机数组。

B = zeros([d1 d2 d3...])命令将同样返回一个指定维数的随机数组。

B = zeros(size(A))命令将返回一个与 A 大小相同的随机数组。

举例：

在 MATLAB 命令窗口中输入 B = zeros(2)，

结果显示为：

```
B =
     0     0
     0     0
```

在 MATLAB 命令窗口中输入 B = zeros(2,3)，

结果显示为：

```
B =
     0     0     0
     0     0     0
```

在 MATLAB 命令窗口中输入：

```
A=[1 2 3; 4 5 6];
B = zeros(size(A))
```

结果显示为：

```
B =
     0     0     0
     0     0     0
```

4.2　特殊变量和常量

1. 最近的输入命令作出的反应

名称：ans

MATLAB 对最近的输入命令作出的反应。

语法：ans

描述：当用户没有指定输出参数时，系统将自动创建变量“ans”作为输出参数。

举例：

在 MATLAB 命令窗口中输入：

A=[1 2;3 4;5 6];

B=ones(3,2);

A+B

结果为

ans =

    2   3
    4   5
    6   7

2. 识别 MATLAB 运行的计算机

名称：computer

识别 MATLAB 运行的计算机。

语法：有如下两种表达形式：

- str = computer
- [str,maxsize] = computer

描述：str = computer 命令将返回一个带有运行 MATLAB 计算机类型的字符串。变量“str”返回的字符串见表 4-1。

表 4-1　　变量“str”返回的字符串

字符串	代表的计算机类型
ALPHA	DEC Alpha
AXP_VMSG	Alpha VMS G_float
AXP_VMSIEEE	Alpha VMS IEEE
HP700	HP 9000/700
IBM_RS	IBM RS6000 workstation
LNX86	Linux Intel
PCWIN	MS-Windows
SGI	Silicon Graphics (R4000)
SGI64	Silicon Graphics (R8000)
SOL2	Solaris 2 SPARC workstation
SUN4	Sun4 SPARC workstation
VAX_VMSD	VAX/VMS D_float
VAX_VMSG	VAX/VMS G_float

[str,maxsize] = computer 命令将返回一个带有运行 MATLAB 计算机类型的字符串，并返回一个指示当前 MATLAB 版本中一个数组中允许的最大元素数目的数字。

举例：

在 MATLAB 命令窗口中输入 str = computer，

结果显示为：

str =

PCWIN

在 MATLAB 命令窗口中输入[str,maxsize] = computer，

结果显示为：

```
str =
PCWIN
maxsize =
  2.1475e+009
```

3. 浮点相对精度

名称：eps

返回浮点相对精度。

语法：eps

描述：eps 命令返回从 1.0 到下一个最大的浮点数的距离。

举例：

在 MATLAB 命令窗口中输入 eps，

结果为：

```
ans =
  2.2204e-016
```

4. 浮点运算

名称：flops

计算浮点运算。

语法：有如下两种表达形式：

- f = flops
- flops(0)

描述：f = flops 返回累计浮点运算的次数。

flops(0)将累计浮点运算的次数重置为 0。

假设 A 和 B 都是 n×n 实数矩阵，表 4-2 列出了几种典型运算的浮点运算次数。

表 4-2　　典型不同运算的浮点运算次数

操作	浮点运算次数
A+B	n^2
A*B	2*n^3
A^100	99*(2*n^3)
lu(A)	(2/3)*n^3

举例：

在 MATLAB 命令窗口中输入：

```
A=[1 2 3;4 5 6;7 8 9];
B=[9 8 7;6 5 4;3 2 1];
flops(0);
A+B;
f = flops
```

结果显示为：

```
f =
     9
```

5. 虚部单位

名称：i

虚部单位。

语法：有如下几种表达形式：

- i
- a+bi
- x+i*y

描述：作为基础虚部单位，i 被用于输入复数。由于 i 是一个函数，所以能够被覆盖并作为一个变量。用户也可以使用 j 作为虚部单位。

举例：

在 MATLAB 命令窗口中输入 i，

结果显示为：

```
ans =
          0 + 1.0000i
```

6. 无穷大

名称：inf

无穷大。

语法：inf

描述：Inf 返回 IEEE 算法的正无穷大。Inf 通常由类似被 0 除这样的操作产生。

举例：

在 MATLAB 命令窗口中输入 1/0，

结果显示为：

```
Warning: Divide by zero.
ans =
   Inf
```

在 MATLAB 命令窗口中输入：

```
1.e1000
```

结果显示为：

```
ans =
   Inf
```

在 MATLAB 命令窗口中输入：

```
log(0)
Warning: Log of zero.
```

结果显示为：

```
ans = -Inf
```

7. 输入参数名

名称：inputname

输入参数名。

语法：inputname(argnum)

描述：inputname(argnum)命令返回工作空间内指定序号的相应变量名。该命令仅能够被用于一个函数体内。

8. 虚部单位

名称：j

虚部单位。

语法：有如下几种表达形式：

- j
- x+yj
- x+j*y

描述：作为基础虚部单位，j 被用于输入复数。由于 j 是一个函数，所以能够被覆盖并作为一个变量。用户也可以使用 i 作为虚部单位。

举例：

在 MATLAB 命令窗口中输入 j，

结果为：

```
ans =
        0 + 1.0000i
```

9. 非数值

名称：NaN

非数值。

语法：NaN

描述：NaN 返回 IEEE 算法中的非数值。除了“~=”，所有包括 NaN 的逻辑操作都将返回逻辑假。因此，语句 NaN ~= NaN 将返回逻辑真，而语句 NaN == NaN 将返回逻辑假。

举例：

如下操作将产生 NaN：(1) 任何关于 NaN 的算术运算；(2) 正负无穷大的加减运算；(3) 正负无穷大的乘法；(4) 除法，例如 0/0 或 Inf/Inf；(5)求余运算，当 x 是无穷大或 y 是 0 时命令 rem(x,y)将产生 NaN。

10. 函数参数的数目

名称：nargin，nargout

返回函数参数的数目。

语法：有如下几种表达形式：

- n = nargin
- n = nargin('fun')
- n = nargout
- n = nargout('fun')

描述：n = nargin 命令将为一个函数返回其输入参数的数目。

n = nargin('fun')命令将返回指定函数的输入参数的数目。

n = nargout 命令将为一个函数返回其输出参数的数目。

n = nargout('fun')命令将返回指定函数的输出参数的数目。

举例：

下面的例子列出了一个名为“myplot”的函数的部分代码，该函数可以接受可选数目的输入和输出函数。

```
function [X,Y] = myplot(fname,lims,npts,angl,subdiv)
% MYPLOT   绘制一个函数
% MYPLOT(fname,lims,npts,angl,subdiv)
%        前两个输入参数是必须的；
 ...
if nargin < 5, subdiv = 20; end
if nargin < 4, angl = 10; end
if nargin < 3, npts = 25; end
 ...
if nargout == 0
      plot(x,y)
else
      X = x;
      Y = y;
end
```

11. 圆周率

名称：pi

返回圆周率。

语法：pi

描述：pi 返回圆周率的浮点近似值。表达式 4*atan(1)和 imag(log(-1))将返回与 pi 命令相同的值。

举例：

在 MATLAB 命令窗口中输入 pi，

结果显示为：

ans = 3.1416。

在 MATLAB 命令窗口中输入 4*atan(1)，

结果显示为：

```
ans =
     3.1416
```

在 MATLAB 命令窗口中输入：

```
sin(2*pi)
```

结果显示为：

```
ans =
 -2.4493e-016
```

这是因为在 MATLAB 中的 pi 命令产生的不是精确的π值，所以该表达式不为 0。

12. 最大正浮点数

名称：realmax

返回 MATLAB 中的最大正浮点数。

语法：n = realmax

描述：n = realmax 返回特定计算机的最大正浮点数。所有大于该数的数将溢出。realmax 函数等价于表达式 pow2(2-eps,maxexp)。

举例：

在 MATLAB 命令窗口中输入 n = realmax，

结果显示为：

```
n =
  1.7977e+308
```

13. 最小的正浮点数

名称：realmin

返回最小的正浮点数。

语法：n = realmin

描述：n=realmin 返回计算机的最小正浮点数。realmax 等价于表达式 pow2(1,minexp)。

举例：

在 MATLAB 命令窗口中输入 n = realmin，

结果显示为：

```
n =
  2.2251e-308
```

14. 返回参数数目

名称：varargin，varargout

传递或返回参数数目。

语法：有如下两种表达形式：

- function varargout = fun(n)
- y = function fun(varargin)

描述：function varargout = fun(n)命令返回指定函数的参数数目。

y = function fun(varargin)命令接收指定函数的参数数目。

varargin 和 varargout 语句仅能够用于一个 MATLAB 函数的内部。

4.3　时间和日期

1.　日历

名称：calendar

返回日历。

语法：有如下几种表达形式：

- c = calendar
- c = calendar(d)
- c = calendar(y,m)

- calendar(...)

描述：c = calendar 用一个 6×7 矩阵返回当前月份的日历。其中日历的第一列为星期日。

c = calendar(d)命令将返回一个指定月份的日历。

c = calendar(y,m)命令将返回一个指定年份的月的日历。

calendar(...)命令将在屏幕上显示日历。

举例：

在 MATLAB 命令窗口中输入 c = calendar(1999,12)，

结果将显示 1999 年 12 月的日历。

```
                     Dec 1999
     S     M     Tu     W     Th     F     S
     0     0     0      1     2      3     4
     5     6     7      8     9     10    11
    12    13    14     15    16     17    18
    19    20    21     22    23     24    25
    26    27    28     29    30     31     0
     0     0     0      0     0      0     0
```

2. 时间

名称：clock

返回当前的时间。

语法：c = clock

描述：c = clock 命令将返回一个包含当前日期和时间的向量。该向量的格式为：

c = [year month day hour minute seconds]

上式中，前五个元素都为整数，表示秒的元素值精确到十分位。

举例：

在 MATLAB 命令窗口中输入：

c = clock

结果将显示当前的时间：

```
c =
  1.0e+003 *
    1.9990    0.0110    0.0300    0.0220    0.0500    0.0051
```

3. CPU 时间

名称：cputime

返回经过的 CPU 时间。

语法：cputime

描述：cputime 命令将返回从 MATLAB 启动后使用的 CPU 时间的总和。

4. 日期

名称：date

返回当前的日期。

语法：str = date

描述：str = date 命令将返回一个包含当前日期、使用 dd-mmm-yyyy 格式的字符串。

举例：

在 MATLAB 命令窗口中输入 str = date，

结果将显示当前的日期。

```
str =
30-Nov-1999
```

5. 连续日期数

名称：datenum

转换成连续日期数。

语法：有如下几种表达形式：

- N = datenum(str)
- N = datenum(str,P)
- N = datenum(Y,M,D)
- N = datenum(Y,M,D,H,MI,S)

描述：datenum 函数将把日期字符串或日期向量转换为连续日期数。

N = datenum(str)命令将把指定字符串转化为连续日期数。

N = datenum(str,P)命令将把指定字符串转化为连续日期数，并假设两个字符表示的年份在以 P 为中心点的 100 年内。

N = datenum(Y,M,D)命令将根据指定的年月日来返回相应的连续日期数。

N = datenum(Y,M,D,H,MI,S)根据指定的年月日、时分秒来返回相应的连续日期数。

举例：

在 MATLAB 命令窗口中输入 N = datenum(1999,10,30)，

结果显示为：

```
N =
      730423
```

6. 日期字符串格式

名称：datestr

日期字符串格式。

语法：有如下两种表达形式：

- str = datestr(D,dateform)
- str = datestr(D,dateform,P)

描述：str = datestr(D,dateform)命令将把由连续日期数 D 构成的数组中的每个元素转换为一个字符串。当日期字符串的年份由两个字符组成时，例如，22-11-73，将被系统认为是以当前年份为中心的一百年内的年份，即 22-11-73 将被认为是 22-11-1973。

str = datestr(D,dateform,P)命令将把由连续日期数 D 构成的数组中的每个元素转换为一个字符串。当日期字符串的年份由两个字符组成时，将被系统认为是以指定年份 P 为中心的一百年内的年份。

上面的两个命令中的参数“dateform”是可选的，该参数的作用是为结果指定日期格式。“dateform”可以是数字或字符串，见表 4-3。当参数“dateform”的值为 0、1、2、6、13、

14、15 和 16 时，其对应的日期格式对于函数 datenum 和 datevec 是有效的；而其他的值对应的日期格式对于函数 datenum 和 datevec 是无效的。

表 4-3　　　　参数 "dateform"

dateform 的数字表示	dateform 的字符串表示
0	'dd-mmm-yyyy HH:MM:SS'
1	'dd-mmm-yyyy'
2	'mm/dd/yy'
3	'mmm'
4	'm'
5	'mm'
6	'mm/dd'
7	'dd'
8	'ddd'
9	'd'
10	'yyyy'
11	'yy'
12	'mmmyy'
13	'HH:MM:SS'
14	'HH:MM:SS PM'
15	'HH:MM'
16	'HH:MM PM'
17	'QQ-YY'
18	'QQ'

7. 显示日期的组成成分

名称：datevec

显示日期的组成成分。

语法：有如下几种表达形式：

- C = datevec(A)
- C = datevec(A,P)
- [Y,M,D,H,MI,S] = datevec(A)

描述：C = datevec(A)把 A 分为一个每行都包含向量[Y,M,D,H,MI,S]的 n×6 数组，向量的前五个元素为整数。作为输入的 A 既可以包含由 datestr 函数产生的字符串；也可以是由 datenum 和 now 函数产生的标量。当日期字符串的年份由两个字符组成时，例如，22-11-73，将被系统认为是以当前年份为中心的一百年内的年份，即 22-11-73 将被认为是 22-11-1973。

C = datevec(A,P) 命令将把 A 分为一个每行都包含向量[Y,M,D,H,MI,S]的 n×6 数组，向量的前五个元素为整数。作为输入的 A 既可以包含由 datestr 函数产生的字符串；也可以是由 datenum 和 now 函数产生的标量。当日期字符串的年份由两个字符组成时，将被系统认为是以指定年份 P 为中心的一百年内的年份。

[Y,M,D,H,MI,S] = datevec(A)命令将把日期向量的成分作为单独的变量返回。

当用户创建自己的日期向量时，不必使日期向量的成分都为整数。任何超出常规范围的成分将影响到更高一级的成分。例如，反常的 9 月 31 日将被改变为 10 月 1 日。

举例：

在 MATLAB 命令窗口中输入：

datevec('11/22/1973')

结果显示为：

ans =

1973　　11　　22　　0　　0　　0

在 MATLAB 命令窗口中输入：

datevec('2/31/1973')

由于 1973 年 2 月份只有 28 天，所以结果显示为：

ans =

1973　　3　　3　　0　　0　　0

8. 月末的日期

名称：eomday

返回月末的日期。

语法：E = eomday(Y,M)

描述：E = eomday(Y,M)命令将返回由相应的数组 Y，M 指定年份的月末的日期。

举例：

在 MATLAB 命令窗口中输入 E = eomday(1988,2)，

结果显示为：

E =

29

因为 1988 年是一个闰年，所以 2 月份有 29 天。

下面一段程序代码可以显示出指定年份段内的所有闰年。其中，n 为指定年份，a 为整数：

```
y =  n : n +a;
E = eomday(y,2*ones(length(y),1)');
y(find(E==29))
```

若希望显示 1860 年到 1960 年区间的所有闰年，只需要将上面的程序代码改为：

```
y = 1860:1960;
E = eomday(y,2*ones(length(y),1)');
y(find(E==29))
```

结果将显示：

```
ans =
  Columns 1 through 6
      1860      1864      1868      1872      1876      1880
  Columns 7 through 12
      1884      1888      1892      1896      1904      1908
  Columns 13 through 18
      1912      1916      1920      1924      1928      1932
  Columns 19 through 24
      1936      1940      1944      1948      1952      1956
  Column 25
      1960
```

9. 经过的时间

名称：etime

返回经过的时间。

语法：e = etime(t2,t1)

描述：e = etime(t2,t1)命令将返回指定的两个 t1、t2 向量间的秒数。t1、t2 必须是由 clock 命令返回的向量，其格式为：T = [Year Month Day Hour Minute Second]。

举例：

在 MATLAB 命令窗口中输入：

```
t1 = clock;
t2 = clock;
e = etime(t2,t1)
```

结果显示为：

```
e =
    19.0600
```

10. 当前的日期和时间

名称：now

返回当前的日期和时间。

语法：t = now

描述：t = now 命令把当前的日期和时间作为一个连续时间数返回到变量 t 中。

如果只需要返回当前时间的连续日期数，可以使用 rem(now,1)命令；如果只需要返回当前日期的连续日期数，可以使用 floor(now)命令。

举例：

在 MATLAB 命令窗口中输入：

```
t1 = now, t2 = rem(now,1), t3 = floor(now)
```

结果显示为：

```
t1 =
   7.3046e+005
t2 =
      0.5970
t3 =
        730455
```

11. 秒表定时器

名称：tic，toc

秒表定时器。

语法：有如下几种表达形式：

- tic
- toc
- t = toc

描述：tic 命令将启动一个秒表定时器。

toc 命令将在屏幕上显示出从秒表定时器被启动后经过的时间。

t = toc 命令将经过的时间返回到变量 t 中。

12. 星期

名称：weekday

显示指定日期为星期几。

语法：[N,S] = weekday(D)

描述：[N,S] = weekday(D)命令将根据指定的连续时间数数组或日期字符串，使用数字 N 和字符串返回该日期位于一个星期中的第几天。N 和 S 的对应值见表 4-4。

表 4-4　　N 和 S 的对应值

N	S
1	Sun
2	Mon
3	Tue
4	Wed
5	Thu
6	Fri
7	Sat

举例：

在 MATLAB 命令窗口中输入：

```
[n,s] = weekday('23-Nov-1976')
```

结果显示为：

```
n =
      3
s =
      Tue
```

4.4 矩阵操作

1. 连接数组

名称：cat

连接数组。

语法：有如下两种表达形式：

- C = cat(dim,A,B)
- C = cat(dim,A1,A2,A3,A4...)

描述：C = cat(dim,A,B)命令将把数组 A 和 B 按照指定的维数“dim”连接起来。

C = cat(dim,A1,A2,A3,A4...)把数组 A1、A2、A3、A4 等按照指定的维数 dim 连接起来。

cat(1,A,B)命令同[A;B]命令是等价的。cat(2,A,B)命令同[A,B]命令是等价的。

举例：

在 MATLAB 命令窗口中输入：

```
A = [1 3;5 7];
```

```
B = [2 4;6 8];
cat(1,A,B)
```

结果显示为：

```
ans =
     1     3
     5     7
     2     4
     6     8
```

继续输入：

```
cat(2,A,B)
```

结果显示为：

```
ans =
     1     3     2     4
     5     7     6     8
```

2. 对角矩阵

名称：diag

对角矩阵。

语法：有如下几种表达形式：

- X = diag(v,k)
- X = diag(v)
- v = diag(X,k)
- v = diag(X)

描述：X = diag(v,k)命令当 v 是一个包含 n 个元素的向量时，返回一个阶数为 n+abs(k)的方阵 X，其第 k 阶对角线上为向量 v 中的元素。

X = diag(v)命令当 v 是一个包含 n 个元素的向量时，返回一个方阵 X，其主对角线上为向量 v 中的元素。

v = diag(X,k)对于矩阵 X，返回一个列向量，该向量由 X 的第 k 阶对角线上的元素构成。

v = diag(X)命令对于矩阵 X，返回一个列向量，该向量由 X 的主对角线上的元素构成。

k=0 表示主对角线，k>0 表示在主对角线之上，k<0 表示在主对角线之下。

举例：

在 MATLAB 命令窗口中输入：

```
v = [2 5 8];
diag(v,0)
```

结果为：

```
ans =
     2     0     0
     0     5     0
     0     0     8
```

继续输入：

```
diag(v,1)
```
结果为：
```
ans =
     0     2     0     0
     0     0     5     0
     0     0     0     8
     0     0     0     0
```
继续输入：
```
diag(v,-1)
```
结果为：
```
ans =
     0     0     0     0
     2     0     0     0
     0     5     0     0
     0     0     8     0
```
在 MATLAB 命令窗口中输入：
```
X = magic(4);
diag(X)
```
结果为：
```
ans =
    16
    11
     6
     1
```
继续输入：
```
diag(X,1)
```
结果为：
```
ans =
     2
    10
    12
```
3. *矩阵做左右翻转*

名称：fliplr

对矩阵做左右翻转。

语法：B = fliplr(A)

描述：B = fliplr(A)命令将返回矩阵 A 左右翻转后的结果。

举例：

在 MATLAB 命令窗口中输入：
```
A = [1 2 3;4 5 6;7 8 9];
```

```
B = fliplr(A)
```

结果显示为：

```
B =
     3     2     1
     6     5     4
     9     8     7
```

4. 矩阵做上下翻转

名称：flipud

对矩阵做上下翻转。

语法：B = flipud(A)

描述：B = flipud(A)命令将返回矩阵 A 上下翻转后的结果。

举例：

在 MATLAB 命令窗口中输入：

```
A = [1 2 3;4 5 6;7 8 9];
B = flipud(A)
```

结果显示为：

```
B =
     7     8     9
     4     5     6
     1     2     3
```

5. 粘贴数组

名称：repmat

复制并粘贴一个数组。

语法：有如下几种表达形式：

- B = repmat(A,m,n)
- B = repmat(A,[m n])
- B = repmat(A,[m n p...])

描述：B = repmat(A,m,n)命令将根据数组 A 创建一个由 A 构成的 m×n 数组 B。

B = repmat(A,[m n])命令的作用与 B = repmat(A,m,n)命令的作用相同。

B = repmat(A,[m n p...])命令将根据数组 A 创建一个由 A 构成的多维(m×n×p...)数组 B。数组 A 本身可以是多维数组。

举例：

在 MATLAB 命令窗口中输入：

```
A = [1 2;3 4];
B=repmat(A,2,1)
```

结果显示为：

```
B =
     1     2
     3     4
```

```
1    2
3    4
```

6. 重新构造数组

名称：reshape

重新构造数组。

语法：有如下几种表达形式：

- B = reshape(A,m,n)
- B = reshape(A,m,n,p,...)
- B = reshape(A,[m n p...])

描述：B = reshape(A,m,n)命令将返回一个m×n矩阵B，且B中的元素是按照列方法从矩阵A中取出的。当矩阵A不是由m×n个元素构成时，该命令将返回一个错误。

B = reshape(A,m,n,p,...)命令将返回一个由数组A中的元素构成的N维数组B，且B中的元素是按照"m×n×p×..."的方式构成的。当矩阵 A 不是由"m×n×p×..."个元素构成时，该命令将返回一个错误。

B = reshape(A,[m n p...]命令的作用与B = reshape(A,m,n,p,...)命令的作用是相同的。

举例：

在MATLAB命令窗口中输入：

```
A = [1 2 3 4;5 6 7 8];
B=reshape(A,4,2)
```

结果显示为：

```
B =

    1    3
    5    7
    2    4
    6    8
```

7. 矩阵旋转90度

名称：rot90

将指定矩阵旋转90度。

语法：有如下两种表达形式：

- B = rot90(A)
- B = rot90(A,k)

描述：B = rot90(A)命令将把矩阵A按逆时针方向旋转90度。

B = rot90(A,k)命令将把矩阵A按逆时针方向旋转90×k度，其中，k为整数。

举例：

在MATLAB命令窗口中输入：

```
A = magic(3);
B = rot90(A)
```

结果显示为

```
B =
```

```
    6     7     2
    1     5     9
    8     3     4
```

8. 下三角矩阵

名称：tril

返回一个矩阵的下三角矩阵。

语法：有如下两种表达形式：

- L = tril(X)
- L = tril(X,k)

描述：L = tril(X)命令将返回矩阵 X 的下三角矩阵。

L = tril(X,k)命令将返回矩阵 X 的第 k 条对角线的下三角矩阵。

当 k=0 时，L = tril(X)命令与 L = tril(X,k)命令作用相同。

举例：

在 MATLAB 命令窗口中，输入：

```
A = ones(3);
B = tril(A,-1)
```

结果显示为：

```
B =
    0     0     0
    1     0     0
    1     1     0
```

9. 上三角矩阵

名称：triu

返回一个矩阵的上三角矩阵。

语法：有如下两种表达形式：

- U = triu(X)
- U = triu(X,k)

描述：U = triu(X)命令将返回矩阵 X 的上三角矩阵。

U = triu(X,k)命令将返回矩阵 X 的第 k 条对角线的上三角矩阵。

举例：

在 MATLAB 命令窗口中输入：

```
A = ones(3);
B = triu(A,1)
```

结果显示为：

```
B =
    0     1     1
    0     0     1
    0     0     0
```

4.5　特殊矩阵函数

1. 伴随矩阵

名称：compan

伴随矩阵。

语法：A = compan(u)

描述：A = compan(u)命令将返回相应的伴随矩阵，该矩阵的第一行为-u(2:n)/u(1)，u 为多项式系数向量，compan(u)的特征值为多项式的根。

举例：

对于某个多项式，其系数为

u = [1　0　-7 -6]

在 MATLAB 命令窗口中输入：

```
A = compan(u)
A =
     0     7     6
     1     0     0
     0     1     0
```

继续输入命令求其特征值：

```
eig(A)
```

结果为：

```
ans =
    3.0000
   -2.0000
   -1.0000
```

该结果说明这个多项式的三个根分别为 3、-2 和-1。

2. 测试矩阵

名称：gallery

测试矩阵。

语法：有如下几种表达形式：

- [A,B,C,...] = gallery('tmfun',P1,P2,...)
- gallery(3)
- gallery(5)

描述：[A,B,C,...] = gallery('tmfun',P1,P2,...)命令将返回由字符串 tmfun 指定的测试矩阵。P1、P2 等是被要求的输入参数。

gallery(3)是一个条件数差的 3×3 矩阵。

gallery(5)是一个有趣的特征值问题。

举例：

在 MATLAB 命令窗口中输入：

X = gallery(3)

结果为：

```
X =
  -149   -50  -154
   537   180   546
   -27    -9   -25
```

继续输入：

X = gallery(5)

结果为：

```
X =
          -9          11         -21          63        -252
          70         -69         141        -421        1684
        -575         575       -1149        3451      -13801
        3891       -3891        7782      -23345       93365
        1024       -1024        2048       -6144       24572
```

3. 哈达马德矩阵

名称：hadamard

哈达马德矩阵。

语法：H = hadamard(n)

描述：H = hadamard(n)命令将根据所给的 n 返回一个哈达马德矩阵。

举例：

在 MATLAB 命令窗口中输入：

H = hadamard(2^2)

结果显示为：

```
H =
     1     1     1     1
     1    -1     1    -1
     1     1    -1    -1
     1    -1    -1     1
```

4. 汉克尔矩阵

名称：hankel

汉克尔矩阵。

语法：有如下两种表达形式：

- H = hankel(c)
- H = hankel(c,r)

描述：H = hankel(c)命令将根据 c 返回一个汉克尔方阵，其第一列为 c，且其反对角线下的元素为 0。

H = hankel(c,r)命令将返回一个第一列为 c、最后一行为 r 的汉克尔矩阵。

举例：

在 MATLAB 命令窗口中输入：

H = hankel(1:5)

结果显示为：

H =

1	2	3	4	5
2	3	4	5	0
3	4	5	0	0
4	5	0	0	0
5	0	0	0	0

在 MATLAB 命令窗口中输入：

H = hankel(1:4,2:10)

结果显示为：

H =

1	2	3	4	3	4	5	6	7
2	3	4	3	4	5	6	7	8
3	4	3	4	5	6	7	8	9
4	3	4	5	6	7	8	9	10

5. 希尔伯特矩阵

名称：hilb

希尔伯特矩阵。

语法：H = hilb(n)

描述：H = hilb(n)命令按照指定的 n 返回一个希尔伯特矩阵。

举例：

在 MATLAB 命令窗口中输入 H = hilb(4)，

结果为：

H =

1.0000	0.5000	0.3333	0.2500
0.5000	0.3333	0.2500	0.2000
0.3333	0.2500	0.2000	0.1667
0.2500	0.2000	0.1667	0.1429

6. 逆希尔伯特矩阵

名称：invhilb

逆希尔伯特矩阵。

语法：H = invhilb(n)

描述：H = invhilb(n)命令将按照指定的 n 返回一个希尔伯特矩阵的逆阵。当 n 小于 15 时，该命令返回的逆阵是精确的；当 n 大于 15 时，该命令返回的逆阵是近似的。

举例：

在 MATLAB 命令窗口中输入 H = invhilb(3)，

结果为：

```
H =
     9   -36    30
   -36   192  -180
    30  -180   180
```

7. 魔术方阵

名称：magic

魔术方阵。

语法：M = magic(n)

描述：M = magic(n)根据指定的 n 产生一个魔术方阵。n 必须是一个大于 3 的标量。

举例：

在 MATLAB 命令窗口中输入 M = magic(4)，

结果为：

```
M =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

8. 帕斯卡矩阵

名称：pascal

帕斯卡矩阵。

语法：A = pascal(n)

描述：A = pascal(n)命令将根据指定的 n 返回一个帕斯卡矩阵。

举例：

在 MATLAB 命令窗口中输入 A = pascal(3)，

结果为：

```
A =
     1     1     1
     1     2     3
     1     3     6
```

9. 托普利茨矩阵

名称：toeplitz

托普利茨矩阵。

语法：有如下两种表达形式：

- T = toeplitz(c,r)
- T = toeplitz(r)

描述：T = toeplitz(c,r)命令将返回以 c 为第一列、r 为第一行的不对称托普利茨矩阵。T = toeplitz(r)命令将根据向量 r 返回对称或厄密共轭托普利茨矩阵。

举例：

在 MATLAB 命令窗口中输入 T = toeplitz(1:4,2:5)，

结果显示为

```
T =
     1     3     4     5
     2     1     3     4
     3     2     1     3
     4     3     2     1
```

10. 威尔金森特征值测试矩阵

名称：wilkinson

威尔金森特征值测试矩阵。

语法：W = wilkinson(n)

描述：W = wilkinson(n)命令将根据指定的 n 返回一个 J. H. Wilkinson 特征值测试矩阵。

举例：

在 MATLAB 命令窗口中输入 W = wilkinson(5)，

结果显示为：

```
W =
     2     1     0     0     0
     1     1     1     0     0
     0     1     0     1     0
     0     0     1     1     1
     0     0     0     1     2
```

第五章　数学函数和坐标变换

5.1　基本数学函数

1.　绝对值

名称：abs

求绝对值函数。

语法：Y = abs(X)

描述：Y = abs(X)返回 X 中每个元素的绝对值。如果 X 为复数，则该命令返回 X 的模。

举例：

在 MATLAB 命令窗口中输入：

```
X = [1 -1 1;-1 1 -1;1 1 1;-1 -1 -1];
Y = abs(X)
```

结果为：

```
Y =
     1     1     1
     1     1     1
     1     1     1
     1     1     1
```

2.　反余弦和反双曲线余弦

名称：acos，acosh

反余弦和反双曲线余弦函数。

语法：有如下两种表达形式：

- Y = acos(X)
- Y = acosh(X)

描述：Y = acos(X)命令将返回 X 中每个元素的反正弦函数值。当 X 中的元素值在[-1,1]区间内时，acos(X)的值为实数且值域在[0,π]区间内；当 X 中的元素值在[-1,1]区间外时，acos(X)的值为复数。

Y = acosh(X) 命令将返回 X 中每个元素的反双曲线余弦函数值。

这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入 acos(-1:0.25:1)，

结果显示为：

```
ans =
  Columns 1 through 7
    3.1416    2.4189    2.0944    1.8235    1.5708    1.3181    1.0472
  Columns 8 through 9
    0.7227         0
```

继续输入：

```
acosh(0:pi/10:pi)
```

结果显示为：

```
ans =
  Columns 1 through 4
        0 + 1.5708i        0 + 1.2512i        0 + 0.8914i        0 + 0.3408i
  Columns 5 through 8
   0.7019             1.0232             1.2478             1.4250
  Columns 9 through 11
   1.8115
```

3.　*反余切和反双曲线余切*

名称：acot，acoth

反余切和反双曲线余切函数。

语法：有如下两种表达形式：

- Y = acot(X)
- Y = acoth(X)

描述：Y = acot(X)命令将返回 X 中每个元素的反余切函数值。

Y = acoth(X)命令将返回 X 中每个元素的反双曲线余切函数值。

这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入：

```
Y = acot(1:pi/8:2*pi)
```

结果为：

```
Y =
  Columns 1 through 7
    0.7854    0.6227    0.5106    0.4304    0.3710    0.3254    0.2896
  Columns 8 through 14
    0.2607    0.2369    0.2171    0.2002    0.1858    0.1733    0.1624
```

继续输入：

```
Y = acoth(1:pi/8:2*pi)
```

结果显示为：

```
Y =
  Columns 1 through 7
```

```
      Inf    0.9036    0.6330    0.4962    0.4106    0.3512    0.3073
  Columns 8 through 14
    0.2734    0.2463    0.2242    0.2058    0.1902    0.1769    0.1653
```

4. 反余割和反双曲线余割

名称：acsc，acsch

反余割和反双曲线余割函数。

语法：有如下两种表达形式：

- Y = acsc(X)
- Y = acsch(X)

描述：Y = acsc(X)命令将返回 X 中每个元素的反余割函数值。

Y = acsch(X)命令将返回 X 中每个元素的反双曲线余割函数值。

这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入 Y = acsc(1:pi/8:2*pi)，

结果为：

```
Y =
  Columns 1 through 7
    1.5708    0.8010    0.5945    0.4770    0.3995    0.3442    0.3026
  Columns 8 through 14
    0.2700    0.2439    0.2224    0.2044    0.1891    0.1760    0.1645
```

继续输入 Y = acsch(1:pi/8:2*pi)，

结果为：

```
Y =
  Columns 1 through 7
    0.8814    0.6674    0.5343    0.4443    0.3798    0.3313    0.2937
  Columns 8 through 14
    0.2637    0.2392    0.2188    0.2016    0.1869    0.1742    0.1631
```

5. 相位角

名称：angle

相位角。

语法：P = angle(Z)

描述：P = angle(Z)命令将返回指定复数数组每个元素的相位角。返回的相位角使用弧度表示，相位角的值在(-π,π)之间。

举例：

在 MATLAB 命令窗口中输入：

```
Z =
   [1.0000 + 1.0000i    1.0000 - 2.0000i    1.0000 + 3.0000i    1.0000 - 4.0000i
    2.0000 - 2.0000i    2.0000 + 2.0000i    2.0000 - 3.0000i    2.0000 + 2.0000i
```

```
    3.0000 + 3.0000i    3.0000 - 2.0000i    3.0000 + 3.0000i    3.0000 - 3.0000i
    4.0000 - 4.0000i    4.0000 + 2.0000i    4.0000 - 3.0000i    4.0000 + 4.0000i]
P = angle(Z)
```

结果为：

```
P =
    0.7854   -1.1071    1.2490   -1.3258
   -0.7854    0.7854   -0.9828    0.7854
    0.7854   -0.5880    0.7854   -0.7854
   -0.7854    0.4636   -0.6435    0.7854
```

6.　反正割和反双曲线正割

名称：asec，asech

反正割和反双曲线正割函数。

语法：有如下两种表达形式：

- Y = asec(X)
- Y = asech(X)

描述：Y = asec(X)命令将返回 X 中每个元素的反正割函数值。

Y = asech(X)命令将返回 X 中每个元素的反双曲线正割函数值。

这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入 Y = asec(1:pi/8:2*pi)，

结果为：

```
Y =
  Columns 1 through 7
         0    0.7698    0.9763    1.0938    1.1713    1.2266    1.2682
  Columns 8 through 14
    1.3008    1.3269    1.3484    1.3664    1.3817    1.3948    1.4063
Y = asech(1:pi/8:2*pi)
```

结果为：

```
Y =
  Columns 1 through 4
        0                   0 + 0.7698i         0 + 0.9763i         0 + 1.0938i
  Columns 5 through 8
        0 + 1.1713i         0 + 1.2266i         0 + 1.2682i         0 + 1.3008i
  Columns 9 through 12
        0 + 1.3269i         0 + 1.3484i         0 + 1.3664i         0 + 1.3817i
  Columns 13 through 14
```

0 + 1.3948i 0 + 1.4063i

7. 反正弦和反双曲线正弦

名称：asin，asinh

反正弦和反双曲线正弦函数。

语法：有如下两种表达形式：

- Y = asin(X)
- Y = asinh(X)

描述：Y = asin(X)命令将返回 X 中每个元素的反正弦函数值。

Y = asinh(X)命令将返回 X 中每个元素的反双曲线正弦函数值。

这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入 Y = asin(-1:.25:1)，

结果为：

```
Y =
  Columns 1 through 7
   -1.5708    -0.8481    -0.5236    -0.2527      0     0.2527     0.5236
  Columns 8 through 9
    0.8481     1.5708
```

继续输入 Y = asinh(1:pi/8:2*pi)，

结果为：

```
Y =
  Columns 1 through 7
    0.8814     1.1337     1.3433     1.5206     1.6732     1.8068     1.9254
  Columns 8 through 14
    2.0319     2.1285     2.2168     2.2980     2.3733     2.4434     2.5089
```

8. 反正切和反双曲线正切

名称：atan，atanh

反正切和反双曲线正切函数。

语法：有如下两种表达形式：

- Y = atan(X)
- Y = atanh(X)

描述：Y = atan(X)命令将返回 X 中每个元素的反正切函数值。

Y = atanh(X)命令将返回 X 中每个元素的反双曲线正切函数值。

这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入 Y = atan(-1:.25:1)，

结果为：

```
Y =
  Columns 1 through 7
   -0.7854    -0.6435    -0.4636    -0.2450         0    0.2450    0.4636
  Columns 8 through 9
    0.6435    0.7854
```

继续输入 Y = atanh(1:pi/8:2*pi)，

结果为：

```
Y =
  Columns 1 through 4
       Inf                  0.9036 + 1.5708i   0.6330 + 1.5708i   0.4962 + 1.5708i
  Columns 5 through 8
   0.4106 + 1.5708i   0.3512 + 1.5708i   0.3073 + 1.5708i   0.2734 + 1.5708i
  Columns 9 through 12
   0.2463 + 1.5708i   0.2242 + 1.5708i   0.2058 + 1.5708i   0.1902 + 1.5708i
  Columns 13 through 14
   0.1769 + 1.5708i   0.1653 + 1.5708i
```

9. 四象限反正切

名称：atan2

四象限反正切函数。

语法：P = atan2(Y,X)

描述：P = atan2(Y,X)命令将返回一个同 X、Y 同维的数组 P，其元素为 X、Y 中元素实部的四象限反正切值，元素的虚部将被忽略。数组 P 中的元素的值域为[-pi,pi]。

举例：

对于任意一个复数 z=x+yi，若将其转化为极坐标，可以使用如下公式：

```
r = abs(z)
theta = atan2(imag(z),real(z))
```

10. 向正无穷大舍入

名称：ceil

向正无穷大舍入。

语法：B = ceil(A)

描述：B = ceil(A)命令将把数组 A 中的元素向最接近的大于或等于该元素的整数舍入。当 A 中的元素是复数时，ceil 命令将分别对该元素的实部和虚部进行操作。

举例：

在 MATLAB 命令窗口中输入：

```
A = [-5 -3.8 -1.2 0 1.2 3.8 5 -1.2-3.8i 1.2+3.8i];
B = ceil(A)
```

结果为：

```
B =
  Columns 1 through 4
```

```
  -5.0000            -3.0000            -1.0000            0
  Columns 5 through 8
   2.0000             4.0000             5.0000            -1.0000 - 3.0000i
  Column 9
   2.0000 + 4.0000i
```

11. 复数

名称：complex

构造复数。

语法：有如下两种表达形式：

- c = complex(a,b)
- c = complex(a)

描述：c = complex(a,b)命令将根据输入的 a、b 构造复数 c。

c = complex(a)命令将把指定的输入 a 作为 c 的实部对待，而 c 的虚部为 0。

输入的 a 和 b 必须同为规模相同的向量、矩阵或多维数组。

举例：

在 MATLAB 命令窗口中输入：

```
a = [1 2;3 4];
b = [5 6;7 8];
c = complex(a,b)
```

结果为

```
c =
    1.0000 + 5.0000i    2.0000 + 6.0000i
    3.0000 + 7.0000i    4.0000 + 8.0000i
```

12. 共轭复数

名称：conj

共轭复数。

语法：ZC = conj(Z)

描述：ZC = conj(Z)命令将返回 Z 中每个元素的共轭复数。如果 Z 是一个复数数组，则：

conj(Z) = real(Z) - i*imag(Z)

举例：

在 MATLAB 命令窗口中输入：

```
a = [1 2;3 4];
b = [5 6;7 8];
c = complex(a,b);
zc = conj(c)
```

结果为：

```
zc =
    1.0000 - 5.0000i    2.0000 - 6.0000i
    3.0000 - 7.0000i    4.0000 - 8.0000i
```

13．余弦和双曲线余弦

名称：cos，cosh

余弦和双曲线余弦函数。

语法：有如下两种表达形式：

- Y = cos(X)
- Y = cosh(X)

描述：Y = cos(X)命令将返回 X 中每个元素的余弦函数值。

Y = cosh(X)命令将返回 X 中每个元素的双曲线余弦函数值。这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入 Y = cos(-1:.25:1)，

结果为：

```
Y =
  Columns 1 through 7
    0.5403    0.7317    0.8776    0.9689    1.0000    0.9689    0.8776
  Columns 8 through 9
    0.7317    0.5403
```

继续输入 Y = cosh(1:pi/8:2*pi)，

结果为

```
Y =
  Columns 1 through 7
    1.5431    2.1371    3.0648    4.4714    6.5764    9.7086   14.3574
  Columns 8 through 14
   21.2488   31.4594   46.5841   68.9855  102.1627  151.2982  224.0672
```

14．余切和双曲线余切

名称：cot，coth

余切和双曲线余切函数。

语法：有如下两种表达形式：

- Y = cot(X)
- Y = coth(X)

描述：Y = cot(X)命令将返回 X 中每个元素的余切函数值。

Y = coth(X)命令将返回 X 中每个元素的双曲线余切函数值。这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入 Y = cot(1:.25:3)，

结果为：

```
Y =
  Columns 1 through 7
    0.6421    0.3323    0.0709   -0.1811   -0.4577   -0.8073   -1.3386
```

```
  Columns 8 through 9
   -2.4218    -7.0153
```

继续输入：

Y = coth(1:pi/12:pi)

结果为：

```
Y =
  Columns 1 through 7
    1.3130    1.1743    1.0997    1.0579    1.0339    1.0199    1.0118
  Columns 8 through 9
    1.0070    1.0041
```

15. 余割和双曲线余割

名称：csc，csch

余割和双曲线余割函数。

语法：有如下两种表达形式：

- Y = csc(X)
- Y = csch(X)

描述：Y = csc(X)命令将返回 X 中每个元素的余割函数值。

Y = csch(X)命令将返回 X 中每个元素的双曲线余割函数值。这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入 Y = csc(1:.25:3)，

结果为

```
Y =
  Columns 1 through 7
    1.1884    1.0538    1.0025    1.0163    1.0998    1.2852    1.6709
  Columns 8 through 9
    2.6201    7.0862
```

继续输入 Y = csch(1:pi/12:pi)，

结果为：

```
Y =
  Columns 1 through 7
    0.8509    0.6156    0.4576    0.3452    0.2626    0.2007    0.1538
  Columns 8 through 9
    0.1181    0.0908
```

16. 指数

名称：exp

指数。

语法：Y = exp(X)

描述：Y = exp(X)返回 X 中各元素的指数。计算矩阵的指数时要使用 expm 命令。

举例：

在 MATLAB 命令窗口中输入：

A = rand(4);

exp(A)

结果为：

ans =

2.5860	2.4383	2.2737	2.5138
1.2600	2.1428	1.5600	2.0922
1.8346	1.5785	1.8505	1.1928
1.6258	1.0187	2.2077	1.5004

17. 向 0 舍入

名称：fix

向 0 舍入。

语法：B = fix(A)

描述：B = fix(A)命令把 A 中的元素向 0 舍入，结果将得到一个整数数组。若 A 中的元素是复数，该命令将分别对复数的实部和虚部进行操作。

举例：

在 MATLAB 命令窗口中输入：

A = [-5 -3.8 -1.2 0 1.2 3.8 5 -1.2-3.8i 1.2+3.8i];

B = fix(A)

结果为：

```
B =
  Columns 1 through 4
  -5.0000          -3.0000          -1.0000                0
  Columns 5 through 8
   1.0000           3.0000           5.0000          -1.0000 - 3.0000i
  Column 9
   1.0000 + 3.0000i
```

18. 向负无穷大舍入

名称：floor

向负无穷大舍入。

语法：B = floor(A)

描述：B = floor(A)命令将把数组 A 中的元素向最接近的小于或等于该元素的整数舍入。当 A 中的元素是复数时，floor 命令将分别对该元素的实部和虚部进行操作。

举例：

在 MATLAB 命令窗口中输入：

A = [-5 -3.8 -1.2 0 1.2 3.8 5 -1.2-3.8i 1.2+3.8i];

B = floor(A)

结果为：

B =

Columns 1 through 4

-5.0000	-4.0000	-2.0000	0

Columns 5 through 8

1.0000	3.0000	5.0000	-2.0000 - 4.0000i

Column 9

1.0000 + 3.0000i

19. 最大公约数

名称：gcd

最大公约数。

语法：有如下两种表达形式：

- G = gcd(A,B)
- [G,C,D] = gcd(A,B)

描述：G = gcd(A,B)命令将整数数组 A 和 B 的相应元素的最大公约数返回到数组 G 中。

[G,C,D] = gcd(A,B)命令将整数数组 A 和 B 的相应元素的最大公约数返回到数组 G 中，并将返回数组 C 和 D，C 和 D 必须满足等式 A(i).*C(i) + B(i).*D(i) = G(i)。这两个数组对于解丢番图方程和计算基本哈米特转换是非常有用的。按惯例，命令 gcd(0,0)将返回值 0。

举例：

在 MATLAB 命令窗口中输入：

```
A = [0 1 6;12 12 18];
B = [0 2 4;6 8 10];
G = gcd(A,B)
```

结果为：

```
G =
     0     1     2
     6     4     2
```

20. 复数的虚部

名称：imag

复数的虚部。

语法：Y = imag(Z)

描述：Y = imag(Z)命令返回指定数组 Z 中每个元素的虚部。

举例：

在 MATLAB 命令窗口中输入：

```
Z = [1+i 1-i;2+2i 2-2i;3+3i 3-3i];
Y = imag(Z)
```

结果为：

```
Y =
     1    -1
     2    -2
```

    3   -3

21. 最小公倍数

名称：lcm

最小公倍数。

语法：L = lcm(A,B)

描述：L = lcm(A,B)返回由整数数组 A 和 B 的相应元素的最小公倍数构成的数组 L。A 和 B 必须包含正整数元素且必须大小相等(允许其中一方为标量)。

举例：

在 MATLAB 命令窗口中输入：

```
A = [2 3 5;1 7 9];
B = [3 6 4;2 5 18];
L = lcm(A,B)
```

结果为：

```
L =
     6     6    20
     2    35    18
```

22. 自然对数

名称：log

自然对数。

语法：Y = log(X)

描述：Y = log(X)命令将返回指定数组 X 中每个元素的自然对数值。该函数的定义域包括复数和负数。

举例：

在 MATLAB 命令窗口中输入：

```
X = [-1 3+4i;1 3-5i];
Y = log(X)
```

结果为：

```
Y =
        0 - 3.1416i     1.6094 + 0.9273i
        0                   1.7632 - 1.0304i
```

23. 以 2 为底的对数

名称：log2

以 2 为底的对数。

语法：有如下两种表达形式：

Y = log2(X)

[F,E] = log2(X)

描述：Y = log2(X)命令将计算指定数组 X 中每个元素的以 2 为底的对数值。

[F,E] = log2(X)命令将返回数组 F 和 E。其中，F 是一个实数数组，且对于实数数组 X 满足等式：X = F.*2.^E；E 是一个整数数组，且对于实数数组 X 满足等式：X = F.*2.^E。该

函数的功能相当于 ANSI C 中的 frexp()函数，或 IEEE 浮点标准函数 logb()。该函数的定义域包括复数和负数。

举例:

在 MATLAB 命令窗口中输入：

```
X = [-1 3+4i;5 3-5i];
[F,E] = log2(X)
```

结果为：

```
F =
   -0.5000    0.7500
    0.6250    0.7500
E =
     1     2
     3     2
```

24. 以 10 为底的对数

名称：log10

以 10 为底的对数。

语法：Y = log10(X)

描述：Y = log10(X)命令将返回指定数组 X 中每个元素的以 10 为底的对数值。该函数的定义域包括复数和负数。

举例:

在 MATLAB 命令窗口中输入：

```
X = [-1 3+4i;5 3-5i];
Y = log10(X)
```

结果为：

```
Y =
        0 - 1.3644i    0.6990 + 0.4027i
        0.6990         0.7657 - 0.4475i
```

25. 模除

名称：mod

模除。

语法：M = mod(X,Y)

描述：M = mod(X,Y)命令将用 M 返回 X 模除 Y 的结果。当 X、Y 同号时，mod(X,Y)命令将和 rem(X,Y)的返回值相同；当设 X、Y 同为正时，有如下等式：mod(-x,y) = rem(-x,y)+y

举例:

在 MATLAB 命令窗口中输入：

```
X = [1 2 3;4 5 6;7 8 9];
Z = mod(X,2)
```

结果为：

```
Z =
```

```
     1     0     1
     0     1     0
     1     0     1
```

26. 二项式系数

名称：nchoosek

二项式系数或全组合。

语法：有如下两种表达形式：

- C = nchoosek(n,k)
- C = nchoosek(v,k)

描述：C = nchoosek(n,k)命令对于非负整数 n 和 k，返回结果 n!/((n-k)!k!)。

C = nchoosek(v,k)命令对于一个长度为 n 的行向量 v，将创建一个矩阵，其行为从 v 的 n 个元素中每次取 k 个的所有可能的组合。该矩阵将包括 n!/((n-k)!k!)行和 k 列。

举例：

在 MATLAB 命令窗口中输入：

```
C = nchoosek(4,3)
```

结果为：

```
C =
     4
```

继续输入：

```
C = nchoosek(1:4,3)
```

结果为：

```
C =
     1     2     3
     1     2     4
     1     3     4
     2     3     4
```

27. 复数的实部

名称：real

复数的实部。

语法：X = real(Z)

描述：X = real(Z)命令将返回指定复数数组 Z 中每个元素的实部。

举例：

在 MATLAB 命令窗口中输入：

```
Z = [1+i -1+i;2+2i -2-2i;3+3i -3+3i];
x = real(Z)
```

结果为：

```
x =
     1    -1
     2    -2
```

3 -3

28. 余数

名称：rem

余数。

语法：R = rem(X,Y)

描述：R = rem(X,Y)命令将返回结果 X-fix(X./Y).*Y。X 和 Y 应为整数。当 X、Y 同号时，rem(X,Y)命令将和 mod(X,Y)的返回值相同；当设 X、Y 同为正时，有如下等式：

mod(-x,y) = rem(-x,y)+y

举例：

在 MATLAB 命令窗口中输入：

```
X = [1 2 3;-1 -2 -3];
Z = rem(X,2)
```

结果为：

```
Z =
     1     0     1
    -1     0    -1
```

29. 向最接近的整数舍入

名称：round

向最接近的整数舍入。

语法：Y = round(X)

描述：Y = round(X)命令将把 X 中的元素向最接近的整数舍入。若 X 中的元素为复数，复数的虚部和实部将分别被执行该操作。

举例：

在 MATLAB 命令窗口中输入：

```
X = [-2.8 -1.2;2.8 1.2];
Y = round(X)
```

结果为：

```
Y =
    -3    -1
     3     1
```

30. 正割和双曲线正割

名称：sec，sech

正割和双曲线正割函数。

语法：有如下两种表达形式：

- Y = sec(X)
- Y = sech(X)

描述：Y = sec(X)命令将返回指定数组 X 中每个元素的正割函数值。

Y = sech(X)命令将返回指定数组 X 中每个元素的双曲线正割函数值。这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入 Y = sec(-1:.25:1)，

结果为：

```
Y =
  Columns 1 through 7
    1.8508    1.3667    1.1395    1.0321    1.0000    1.0321    1.1395
  Columns 8 through 9
    1.3667    1.8508
```

继续输入：

```
Y = sech(-1:.25:1)
Y =
  Columns 1 through 7
    0.6481    0.7724    0.8868    0.9695    1.0000    0.9695    0.8868
  Columns 8 through 9
    0.7724    0.6481
```

31. 正负号

名称：sign

正负号函数。

语法：Y = sign(X)

描述：Y = sign(X)命令将返回一个与 X 同样大小的数组 Y。Y 的值见表 5-1。

表 5-1　　数组 Y 的值

条件	值
如果 X 中的相应元素大于 0	1
如果 X 中的相应元素等于 0	0
如果 X 中的相应元素小于 0	-1

举例：

在 MATLAB 命令窗口中输入：

```
X = [1 2 3;0 0 0;-1 -2 -3];
Y = sign(X)
```

结果为：

```
Y =
     1     1     1
     0     0     0
    -1    -1    -1
```

32. 正弦和双曲线正弦

名称：sin，sinh

正弦和双曲线正弦函数。

语法：有如下两种表达形式：

- Y = sin(X)

- Y = sinh(X)

描述：Y = sin(X)命令将返回指定数组 X 中每个元素的正弦函数值。

Y = sinh(X)命令将返回指定数组 X 中每个元素的双曲线正弦函数值。这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入 Y = sin(-2*pi:pi/4:2*pi)，

结果为：

```
Y =
  Columns 1 through 7
    0.0000    0.7071    1.0000    0.7071   -0.0000   -0.7071   -1.0000
  Columns 8 through 14
   -0.7071         0    0.7071    1.0000    0.7071    0.0000   -0.7071
  Columns 15 through 17
   -1.0000   -0.7071   -0.0000
```

继续输入 Y = sinh(-2*pi:pi/4:2*pi)，

结果为：

```
Y =
  Columns 1 through 7
 -267.7449 -122.0735  -55.6544  -25.3672  -11.5487   -5.2280   -2.3013
  Columns 8 through 14
   -0.8687         0    0.8687    2.3013    5.2280   11.5487   25.3672
  Columns 15 through 17
   55.6544  122.0735  267.7449
```

33. 平方根

名称：sqrt

平方根。

语法：B = sqrt(A)

描述：B = sqrt(A)命令将返回数组 A 中元素的每个平方根。

举例：

在 MATLAB 命令窗口中输入：

```
A = [1 -2 3;4 -5 6;7 -8 9];
B = sqrt(A)
```

结果为：

```
B =
   1.0000          0 + 1.4142i        1.7321
   2.0000          0 + 2.2361i        2.4495
   2.6458          0 + 2.8284I        3.0000
```

34. 正切和双曲线正切

名称：tan，tanh

正切和双曲线正切函数。

语法：有如下两种表达形式：

- Y = tan(X)
- Y = tanh(X)

描述：Y = tan(X)命令将返回指定数组 X 中每个元素的正切函数值。

Y = tanh(X)命令将返回指定数组 X 中每个元素的双曲线正切函数值。这两个函数操作数组中的每个元素。这两个函数的定义域和值域都可以包括复数值，所有的角都是弧度。

举例：

在 MATLAB 命令窗口中输入 Y = tan(1:.125:2)，

结果为

```
Y =
  Columns 1 through 7
    1.5574    2.0926    3.0096    5.0419   14.1014  -18.4309   -5.5204
  Columns 8 through 9
   -3.1852   -2.1850
```

继续输入 Y = tanh(-2*pi:pi/4:2*pi)，

结果为：

```
Y =
  Columns 1 through 7
   -1.0000   -1.0000   -0.9998   -0.9992   -0.9963   -0.9822   -0.9172
  Columns 8 through 14
   -0.6558         0    0.6558    0.9172    0.9822    0.9963    0.9992
  Columns 15 through 17
    0.9998    1.0000    1.0000
```

5.2 特殊数学函数

1. 艾利函数

名称：airy

艾利函数。

语法：有如下几种表达形式：

- W = airy(Z)
- W = airy(k,Z)
- [W,ierr] = airy(k,Z)

描述：艾利函数和改良型贝塞尔函数的关系如下：

$$Ai(Z) = \left[\frac{1}{\pi}\sqrt{Z/3}\right] K_{1/3}(\zeta)$$

$$Bi(Z) = \sqrt{Z/3}\,[I_{-1/3}(\zeta) + I_{1/3}(\zeta)]$$

这里，$\zeta = \frac{2}{3}Z^{3/2}$

W = airy(Z)命令将返回复数数组 Z 中每个元素的艾利函数值。

W = airy(k,Z)命令将根据不同的 k 值返回不同的结果。k 值见表 5-2。

表 5-2 参数 k 的值

k 值	返回结果
0	airy(z)
1	Ai'(z)
2	Bi(z)
3	Bi'(z)

[W,ierr] = airy(k,Z)命令将返回一个错误标志数组。ierr 的值见表 5-3。

表 5-3 参数 ierr 的值

ieer 值	说明
1	非法的参数
2	溢出，返回 Inf
3	参数约简时有精度损失
4	不可接受的精度损失
5	不收敛，返回 NaN

举例：

在 MATLAB 命令窗口中输入：

```
Z = [-1+2i 2-3i;-3+4i 4-5i];
W = airy(Z)
```

结果为：

```
W =
   1.0e+002 *
    0.0170 - 0.0142i    0.0001 - 0.0013i
    2.0773 + 2.0461i   -0.0000 - 0.0002I
```

继续输入：

```
W = airy(1,Z)
```

结果为：

```
W =
   1.0e+002 *
   -0.0287 - 0.0087i    0.0010 + 0.0023i
    1.9960 - 6.0468i    0.0003 + 0.0003i
```

2. 第三类贝塞尔函数

名称： besselh

第三类贝塞尔函数(汉克尔函数)。

语法： 有如下几种表达形式：

- H = besselh(nu,K,Z)
- H = besselh(nu,Z)

- [H,ierr] = besselh(...)

描述：如下方程

$$z^2\frac{d^2y}{dz^2}+z\frac{dy}{dz}+(z^2-v^2)y=0$$

被称为贝塞尔方程，这里 v 是一个非负常量。该方程的解即为贝塞尔函数。

对于非负数 v，$J_v(z)$ 和 $J_{-v}(z)$ 构成贝塞尔方程的基本解集。

$Y_v(z)$ 是线性独立于 $J_v(z)$ 的第二个解，其定义如下：

$$Y_v(z)=\frac{J_v(z)\cos(v\pi)-J_{-v}(z)}{\sin(v\pi)}$$

H = besselh(nu,K,Z)命令将根据 K=1 或 K=2 计算复数数组 Z 中每个元素的汉克尔函数。如果 nu 和 Z 是同维数组，则 H 也是与 nu 和 Z 同维的数组。

H = besselh(nu,Z)命令将根据 K=1 计算复数数组 Z 中每个元素的汉克尔函数。

[H,ierr] = besselh(...)命令将返回一个错误标志数组。ierr 的值见表 5-3。

举例：

在 MATLAB 命令窗口中输入：

```
format long
z = (0:0.5:3)';
besselh(1,z)
```

结果为：

```
ans =
                  NaN -                 NaNi
   0.24226845767487 - 1.47147239267024i
   0.44005058574493 - 0.78121282130029i
   0.55793650791010 - 0.41230862697391i
   0.57672480775687 - 0.10703243154094i
   0.49709410246427 + 0.14591813796679i
   0.33905895852594 + 0.32467442479180i
```

3.　改良型贝塞尔函数

名称：besseli，besselk

改良型贝塞尔函数。

语法：有如下几种表达形式：

- I = besseli(nu,Z)
- K = besselk(nu,Z)
- I = besseli(nu,Z,1)
- K = besselk(nu,Z,1)
- [I,ierr] = besseli(...)

- [K,ierr] = besselk(...)

描述：

如下的微分方程：

$$z^2\frac{d^2y}{dz^2} + z\frac{dy}{dz} - (z^2+v^2)y = 0$$

被称为改良型贝塞尔方程，这里 v 是实数常量，该方程的解即为改良型贝塞尔函数。

对于非整数 v，$I_v(z)$ 和 $I_{-v}(z)$ 构成改良型贝塞尔方程的基本解集。$K_v(z)$ 是独立于 $I_v(z)$ 的第二个解。$K_v(z)$ 和 $I_v(z)$ 的定义如下：

$$I_v(z) = (\frac{z}{2})^v \sum_{k=0}^{\infty} \frac{(\frac{z^2}{4})^k}{k!\Gamma(v+k+1)}, \quad K_v(z) = (\frac{\pi}{2})\frac{I_{-v}(z) - I_v(z)}{\sin(v\pi)}$$

I = besseli(nu,Z)命令计算改良型第一类贝塞尔函数。

K = besselk(nu,Z)命令计算改良型第二类贝塞尔函数。

I = besseli(nu,Z,1)命令计算 besseli(nu,Z).*exp(-real(Z))。

K = besselk(nu,Z,1)命令计算 besselk(nu,Z).*exp(real(Z))。

[I,ierr] = besseli(...)命令将返回一个错误标志数组。

[K,ierr] = besselk(...)命令将返回一个错误标志数组。ierr 的值见表 5-3。

举例：

在 MATLAB 命令窗口中输入：

```
format long
z = (0:0.5:3)';
besseli(1,z)
```

结果为：

```
ans =

                  0
   0.25789430539090
   0.56515910399249
   0.98166642857791
   1.59063685463733
   2.51671624528870
   3.95337021740261
```

继续输入：

```
besselk(1,z)
ans =

                Inf
   1.65644112000330
```

0.60190723019723
0.27738780045684
0.13986588181652
0.07389081634775
0.04015643112819

4. 贝塞尔函数

名称：besselj，bessely

贝塞尔函数。

语法：有如下几种表达形式：

- J = besselj(nu,Z)
- Y = bessely(nu,Z)
- J = besselj(nu,Z,1)
- Y = bessely(nu,Z,1)
- [J,ierr] = besselj(nu,Z)
- [Y,ierr] = bessely(nu,Z)

描述：微分方程：

$$z^2\frac{d^2 y}{dz^2}+z\frac{dy}{dz}+(z^2-v^2)y=0$$

被称为贝塞尔方程，这里 v 是一个实数常量，该方程的解即为贝塞尔函数。对于非整数 v，$J_v(z)$ 和 $J_{-v}(z)$ 构成贝塞尔方程的基本解集。$J_v(z)$ 的定义如下：

$$J_v(z)=(\frac{z}{2})^v\sum_{k=0}^{\infty}\frac{(-\frac{z^2}{4})^k}{k!\Gamma(v+k+1)}$$，其中，$\Gamma(a)$ 是一个 gamma 函数。

$Y_v(z)$ 是线性独立于 $J_v(z)$ 的第二个解，其定义如下：

$$Y_v(z)=\frac{J_v(z)\cos(v\pi)-J_{-v}(z)}{\sin(v\pi)}$$

J = besselj(nu,Z)命令计算第一类贝塞尔函数。

Y = bessely(nu,Z)命令计算第二类贝塞尔函数。

J = besselj(nu,Z,1)命令计算 besselj(nu,Z).*exp(-imag(Z))。

Y = bessely(nu,Z,1)命令计算 bessely(nu,Z).*exp(-imag(Z))。

[J,ierr] = besselj(nu,Z)命令将返回一个错误标志数组。

[Y,ierr] = bessely(nu,Z)命令将返回一个错误标志数组。

举例：

在 MATLAB 命令窗口中输入：

```
z = (0:0.5:3)';
besselj(1,z)
```

结果为：

```
ans =
                         0
    0.24226845767487
    0.44005058574493
    0.55793650791010
    0.57672480775687
    0.49709410246427
    0.33905895852594
```

继续输入 bessely(1,z)，

结果为：

```
ans =
                      -Inf
   -1.47147239267024
   -0.78121282130029
   -0.41230862697391
   -0.10703243154094
    0.14591813796679
    0.32467442479180
```

5. 贝塔函数

名称：beta，betainc，betaln

贝塔函数。

语法：有如下几种表达形式：

- B = beta(Z,W)
- I = betainc(X,Z,W)
- L = betaln(Z,W)

描述：贝塔函数是：

$$B(z,w) = \int_0^1 t^{z-1}(1-t)^{w-1}\,\mathrm{d}t = \frac{\Gamma(z)\Gamma(w)}{\Gamma(z+w)}$$

这里的 $\Gamma(z)$ 是一个 gamma 函数。

非完全贝塔函数为：

$$I_x(z,w) = \frac{1}{B(z,w)}\int_0^x t^{z-1}(1-t)^{w-1}\,\mathrm{d}t$$

B = beta(Z,W)命令将计算复数数组 Z 和 W 相应元素的贝塔函数。

I = betainc(X,Z,W)命令计算非完全贝塔函数。其中 X 中的元素必须属于区间[0,1]。

L = betaln(Z,W)命令将计算自然对数贝塔函数。

举例：

在 MATLAB 命令窗口中输入：

```
format rat
beta((1:9)',4)
```

结果为：

```
ans =
      1/4
      1/20
      1/60
      1/140
      1/280
      1/504
      1/840
      1/1320
      1/1980
```

继续输入 betainc((.1:.1:.5)',2,2)，

结果为：

```
ans =
      7/250
      13/125
      27/125
      44/125
      1/2
```

继续输入 betaln((1:5)',(2:6)')，

结果为：

```
ans =
 -1588/2291
 -1317/530
 -15710/3837
 -6835/1213
 -8945/1253
```

6. 雅可比椭圆函数

名称：ellipj

雅可比椭圆函数。

语法：有如下两种表达形式：

- [SN,CN,DN] = ellipj(U,M)
- [SN,CN,DN] = ellipj(U,M,tol)

描述：雅可比椭圆函数定义如下：

$$u = \int_0^{\phi} \frac{d\theta}{(1-m\sin^2\theta)^{\frac{1}{2}}}$$

则$sn(u) = \sin\phi,\ cn(u) = \cos\phi,\ dn(u) = (1-\sin^2\phi)^{\frac{1}{2}},\ am(u) = \phi$

一些椭圆函数的定义使用模数 k 替代参数 m，模数 k 和参数 m 的关系为 m=k^2。

[SN,CN,DN] = ellipj(U,M)命令返回雅可比椭圆函数 SN、CN、DN 的值，求其在自变量 U 和参数 M 时的值。

[SN,CN,DN] = ellipj(U,M,tol)命令将按照指定的精度 tol 计算雅可比椭圆函数，默认的精度是 EPS。

7. 完全椭圆积分

名称：ellipke

完全椭圆积分。

语法：有如下几种表达形式：

- K = ellipke(M)
- [K,E] = ellipke(M)
- [K,E] = ellipke(M,tol)

描述：第一类完全椭圆积分的定义如下：

$$K(m) = F(\pi/2|m),$$

其中 F 为第一类椭圆积分，即：

$$K(m) = \int_0^1 [(1-t^2)(1-mt^2)]^{\frac{-1}{2}}\,dt = \int_0^{\frac{\pi}{2}} (1-m\sin^2\theta)^{\frac{-1}{2}}\,d\theta$$

第二类完全椭圆积分的定义如下：

$$E(m) = E(K(m)) = E\langle\pi/2|m\rangle,$$

即：

$$E(m) = \int_0^1 (1-t^2)^{\frac{-1}{2}}(1-mt^2)^{\frac{1}{2}}\,dt = \int_0^{\frac{\pi}{2}} (1-m\sin^2\theta)^{\frac{1}{2}}\,d\theta$$

K = ellipke(M)命令将返回 M 中所有元素的第一类完全椭圆积分。

[K,E] = ellipke(M)命令将返回 M 中所有元素的第一类和第二类完全椭圆积分。

[K,E] = ellipke(M,tol)命令将按照指定的精度 tol 计算第一类和第二类完全椭圆积分，默认的精度是 EPS。

一些椭圆函数的定义使用模数 k 替代参数 m，模数 k 和参数 m 的关系为 m=k^2。

8. 误差函数

名称：erf，erfc，erfcx，erfinv

误差函数。

语法：有如下几种表达形式：

- Y = erf(X)
- Y = erfc(X)

- Y = erfcx(X)
- X = erfinv(Y)

描述：误差函数的定义如下：

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt$$

互补误差函数的定义如下：

$$erfc(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}\,dt = 1 - erf(x)$$

比例互补误差函数的定义如下：

$$erfcx(x) = e^{x^2}erfc(x)$$

Y = erf(X)命令将返回实数数组 X 中每个元素的误差函数值。

Y = erfc(X)命令将计算实数数组 X 中每个元素的互补误差函数值。

Y = erfcx(X)命令将计算实数数组 X 中每个元素的比例互补误差函数值。

X = erfinv(Y)返回 Y 中每个元素的逆误差函数值，Y 中的元素必须在区间(-1,1)内。

误差函数和标准正态概率分布的关系为 standard_normal_cdf = (1 + (erf(x/sqrt(2))))./2。

举例：

在 MATLAB 命令窗口中输入：

```
X = [-1:.5:1]';
Y = erf(X)
```

结果为：

```
Y =
   -0.8427
   -0.5205
         0
    0.5205
    0.8427
```

继续输入：

```
Y = erfc(X)
```

结果为：

```
Y =
    1.8427
    1.5205
    1.0000
    0.4795
    0.1573
```

继续输入：

```
Y = erfcx(X)
```

结果为：

```
Y =
```

5.0090
1.9524
1.0000
0.6157
0.4276

继续输入：

Y = erfinv(X)

结果为：

```
Y =
    -Inf
    -0.4769
    0
    0.4769
    Inf
```

9. 指数积分

名称：expint

指数积分。

语法：Y = expint(X)

描述：指数积分的定义是：

$$\int_x^{\infty} \frac{e^{-t}}{t} \mathrm{d}t$$

指数积分函数的另一个普通定义是柯西主值积分：

$$E_i(x) = \int_{-\infty}^{x} e^{-t} \mathrm{d}t$$

Y = expint(X)命令将求 X 中每个元素的指数积分值。

10. 阶乘函数

名称：factorial

阶乘函数。

语法：factorial(n)

描述：factorial(n)计算 n 的阶乘。因为双精度数仅有 15 位数字，仅当 n≤21 时，使用该命令得到的计算结果才是精确的；当 n>21 时，使用该命令得到的计算结果只有前 15 位是精确的。

举例：

在 MATLAB 命令窗口中输入：

```
format long e
factorial(21)
```

结果为：

```
ans =
```

5.109094217170944e+019

继续输入 factorial(171)。

结果为：

ans = Inf。

11. 伽玛函数

名称：gamma，gammainc，gammaln

伽玛函数。

语法：有如下几种表达形式：

- Y = gamma(A)
- Y = gammainc(X,A)
- Y = gammaln(A)

描述：伽玛函数的定义如下：

$$\Gamma(a) = \int_0^{\infty} e^{-t} t^{a-1} \mathrm{d}t$$

非伽玛函数的定义如下：

$$P(x,a) = \frac{1}{\Gamma(a)} \int_0^x e^{-t} t^{a-1} \mathrm{d}t$$

Y = gamma(A)命令将在实数数组 A 中所有元素上计算伽玛函数。

Y = gammainc(X,A)命令将在数数组 A 中所有元素上计算伽玛函数，A 必须为实数标量。

Y = gammaln(A)命令将在实数数组 A 中所有元素上计算伽玛函数的对数。

使用 gammaln(A)命令时应注意避免上溢和下溢。

12. 勒让德函数

名称：legendre

勒让德函数。

语法：P = legendre(n,X)

描述：勒让德函数的定义为：

$$P_n^m(x) = (-1)^m (1-x^2)^{m/2} \frac{d^m}{dx^m} P_n(x)$$

这里，$P_n(x)$是度 n 的勒让德多项式：

$$P_n(x) = \frac{1}{2^n n!} \left[\frac{\mathrm{d}^n}{\mathrm{d}x} (x^2 - 1)^n \right]$$

P = legendre(n,X)命令在 X 上计算 n 的勒让德函数。参数 n 必须为小于 256 的正整数，X 中元素的定义域必须在区域(-1,1)内。

该命令返回的数组 P 比 X 有更多的维数。

若 X 为一个向量，则 P 将是一个如下格式的矩阵：

$P_2^0(x(1))$ $P_2^0(x(2))$ $P_2^0(x(3))$ ...

$P_2^1(x(1))$ $P_2^1(x(2))$ $P_2^1(x(3))$ ...

$P_2^2(x(1))$ $P_2^2(x(2))$ $P_2^2(x(3))$ ...

举例:

在 MATLAB 命令窗口中输入:

```
X = rand(2,3);
N = 2;
P = legendre(N,X)
```

结果为:

```
P(:,:,1) =
    0.4476    0.8733
   -1.4470   -0.8342
    1.1048    0.2534
P(:,:,2) =
   -0.0903    0.6620
   -1.3367   -1.2534
    2.1807    0.6760
P(:,:,3) =
   -0.4551    0.9399
   -0.5110   -0.5886
    2.9103    0.1203
```

13. 比例浮点数

名称: pow2

比例浮点数。

语法: 有如下两种表达形式:

- X = pow2(Y)
- X = pow2(F,E)

描述: X = pow2(Y)命令计算 2 的 Y 次幂。

X = pow2(F,E)命令对于实数数组 F 和整数数组 E 中的相应元素，计算 x=f*2^e。该函数相当于 ANSI C 函数 ldexp()和 IEEE 浮点标准函数 scalbn()。

举例:

在 MATLAB 命令窗口中输入:

```
F = rand(2,2);
E = [-1 0;1 2];
X = pow2(F,E)
```

结果为:

```
X =
```

0.0683　　0.8939

0.0235　　0.7966

14. 有理逼近

名称：rat，rats

有理逼近。

语法：有如下几种表达形式：

- [N,D] = rat(X)
- [N,D] = rat(X,tol)
- rat(X)
- S = rats(X,strlen)
- S = rats(X)

描述：[N,D] = rat(X)返回两个整数矩阵 N 和 D，在缺省 tol=1.e-6*norm(X(:),1)内 N./D 逼近 X。

[N,D] = rat(X,tol)命令将返回两个整数矩阵 N 和 D，在指定 tol 内 N./D 逼近 X。

rat(X)命令没有输出参数，将简单地显示连续分数。

S = rats(X,strlen)命令将返回包含对 X 中元素简单有理逼近的字符串。参数 strlen 是每个元素的字符串长度，strlen 的默认值等于 13。

S = rats(X)命令将返回与 format rat 命令相同的显示结果。

S = rats(X,strlen)命令中，星号“*”用于分配空间中不能被显示的元素，但不能忽略与 X 中其他元素的比较。

举例：

在 MATLAB 命令窗口中输入 rat(pi)，

结果为：

```
ans =
     3 + 1/(7 + 1/(16))
```

继续输入：

```
[N,D] = rat(pi)
```

结果为

```
N =
   355
D =
   113
```

继续输入：

```
S = rats(rand(4,3),10)
```

结果为：

```
S =
   41/68       174/199      100/101
    9/179        1/67        71/90
   27/65        43/56        25/57
```

18/59　　100/103　　148/297

5.3　坐标转换

1. 笛卡儿坐标转换为极坐标或圆柱坐标

名称：cart2pol

把笛卡儿坐标转换为极坐标或圆柱坐标。

语法：该函数有如下两种表达形式：

- [THETA, RHO, Z] = cart2pol (X, Y, Z)
- [THETA, RHO] = cart2pol (X, Y)

描述：[THETA, RHO, Z] = cart2pol (X, Y, Z)把存储在数组 X(存放各个点的 x 轴坐标)，数组 Y(存放各个点的 y 轴坐标)和数组 Z(存放各个点的 z 轴坐标)中的各个点在三维笛卡儿坐标系中的坐标值转换为对应的圆柱坐标值。其中数组 THETA 中存储的是以 x 轴正向为起点、按逆时针方向旋转的角位移，单位为弧度。数组 RHO 中存储的是坐标原点与各空间点在 x-y 平面内投影点之间的距离。数组 Z 中存储的是空间各点垂直于 x-y 平面的高度。而且要求数组 X， Y，Z 具有相同的维数(或者均为标量，即数组 X，Y 和 Z 中分别只有一个元素)。

[THETA, RHO] = cart2pol(X,Y)把存储在数组 X(存放各个点的 x 轴坐标)和数组 Y(存放各个点的 y 轴坐标)中的各个点在二维笛卡儿坐标系中的坐标值转换为对应的极坐标值。

举例：

(1) 对于三维空间中的五个点，其笛卡儿坐标用数组可表示为

```
X = [1   2   -1   4   0]
Y = [1  -3   -1  -1   8]
Z = [1   1    7  -6   4]
```

在 MATLAB 命令窗口中输入：

```
[r s t] = cart2pol (X, Y, Z)
```

计算结果为：

```
r =
    0.7854   -0.9828   -2.3562   -0.2450    1.5708
s =
    1.4142    3.6056    1.4142    4.1231    8.0000
t =
     1     1     7    -6     4
```

(2) 对于二维空间中的四个点，其笛卡儿坐标用数组可表示为

```
X = [1   1   4   5]
Y = [-1  1  -6   6]
```

在 MATLAB 命令窗口中输入：

```
[r s] = cart2pol (X, Y)
```

计算结果为：

```
r =
    -0.7854    0.7854   -0.9828    0.8761
s =
     1.4142    1.4142    7.2111    7.8102
```

2. 笛卡儿坐标转换为球坐标

名称：cart2sph

把笛卡儿坐标转换为球坐标。

语法：[THETA, PHI, R] = cart2sph (X, Y, Z)

描述：[THETA, PHI, R] = cart2sph (X, Y, Z)把存储在数组 X(存放各个点的 x 轴坐标)，数组 Y(存放各个点的 y 轴坐标)和数组 Z(存放各个点的 z 轴坐标)中的各个点在三维笛卡儿坐标系中的坐标值转换为对应的球坐标值。其中数组 THETA 中存储的是以 x 轴正向为起点、按逆时针方向旋转的角位移(方位角)，单位为弧度。数组 PHI 中存储的是以 x-y 平面为起点、按逆时针方向旋转的角位移(仰角)，单位为弧度。数组 R 中存储的是坐标原点与各个空间点之间的距离。而且要求数组 X, Y, Z 具有相同的维数(或者均为标量，即数组 X, Y, Z 中分别只有一个元素)。

举例：

对于三维空间中的四个点，其笛卡儿坐标用数组可表示为

```
X = [1    3    0    4]
Y = [-1   -2   -6    3]
Z = [1    7    5    2]
```

在 MATLAB 命令窗口中输入：

```
[r s t] = cart2sph (X, Y, Z)
```

计算结果为：

```
r =
    -0.7854   -0.5880   -1.5708    0.6435
s =
     0.6155    1.0952    0.6947    0.3805
t =
     1.7321    7.8740    7.8102    5.3852
```

3. 极坐标或圆柱坐标转换为笛卡儿坐标

名称：pol2cart

把极坐标或圆柱坐标转换为笛卡儿坐标。

语法：该函数有如下两种表达形式：

- [X, Y] = pol2cart(THETA, RHO)
- [X, Y, Z] = pol2cart(THETA, RHO, Z)

描述：[X, Y, Z] = pol2cart(THETA, RHO, Z)把存储在数组 THETA (存放各个空间点在圆柱坐标系中的转角坐标值)，数组 RHO (存放各个空间点在圆柱坐标系中的径向坐标值)和数组 Z(存放各个空间点在圆柱坐标系中的 z 轴坐标值)中的各个点在圆柱坐标系中的坐标值转

换为对应的三维笛卡儿(或 *xyz*)坐标值。其中数组 THETA、RHO 和 Z 要有相同的维数(或者均为标量，即数组 THETA、RHO 和 Z 中分别只有一个元素)，而且 THETA 的单位为弧度。

[X, Y] = pol2cart(THETA, RHO)把存储在数组 THETA (存放各点在极坐标系中的转角坐标值)和数组 RHO(存放各点在极坐标系中的径向坐标值)中的各个点在极坐标系中的坐标值转换为对应的二维笛卡儿(或 *xy*)坐标值。

举例：

(1) 对于圆柱坐标系中的四个三维空间点，其坐标用数组可表示为

```
R = [0.78   1.21   3.14   0.68]
S = [1      7      -4     6]
T = [2      3      -5     3]
```

在 MATLAB 命令窗口中输入：

```
[X Y Z] = pol2cart (R, S, T)
```

计算结果为：

```
X =
    0.4214    0.9122   -2.0524    0.6529
Y =
    0.6563    0.7950    2.3764   -0.1900
Z =
     2     3    -5     3
```

(2) 对于极坐标系中的五个二维空间内的点，其坐标用数组可表示为

```
R = [1.32   3.14   0.245   3.09   -0.87]
S = [1      -2     5       -11    4]
```

在 MATLAB 命令窗口中输入：

```
[X Y] = pol2cart (R, S)
```

计算结果为：

```
X =
    0.2482    2.0000    4.8507   10.9854    2.5793
Y =
    0.9687   -0.0032    1.2128   -0.5673   -3.0573
```

4. 球坐标转换为笛卡儿坐标

名称： sph2cart

把球坐标转换为笛卡儿坐标。

语法： [x, y, z] = sph2cart(THETA, PHI, R)

描述： [x, y, z] = sph2cart(THETA, PHI, R)把存储在数组 THETA (存放各个空间点在球坐标系中的方位角坐标值)，数组 PHI(存放各个空间点在球坐标系中的仰角坐标值)和数组 R(存放各个空间点的径向坐标值)中的各个点在球坐标系中的坐标值转换为对应的三维笛卡儿坐标系中的坐标值。其中数组 THETA 中存储的是以 *x* 轴正向为起点、按逆时针方向旋转的角位移(方位角)，单位为弧度。数组 PHI 中存储的是以 *x-y* 平面为起点、按逆时针方向旋转的

角位移(仰角)，单位为弧度。数组 R 中存储的是坐标原点与各个空间点之间的距离。而且要求数组 THETA, PHI, R 具有相同的维数(或者均为标量，即数组 THETA, PHI, R 中分别只有一个元素)。

举例：

对于球坐标系中的五个三维空间点，其坐标用数组可表示为

```
S = [1.21    0.37    1.06    -4.98   3.32]
R = [-1      -1      5       3       4]
T = [ 4      7       -5      2       -7]
```

在 MATLAB 命令窗口中输入：

```
[X Y Z] = sph2cart (R, S, T)
```

计算结果为：

```
X =
     0.7629    3.5262   -0.6934   -0.5236   -4.5029
Y =
    -1.1882   -5.4917    2.3440    0.0746   -5.2135
Z =
     3.7425    2.5313   -4.3618    1.9288    1.2422
```

第六章　矩阵函数—数值线性代数

6.1　矩阵分析

1. 逆相关条件数

名称：cond

和逆相关的条件数。

语法：该函数有如下两种表达形式：

- c = cond (X)
- c = cond (X, p)

描述：一个矩阵的条件数描述了线性方程组的求解结果对事先给定数据的误差的灵敏度。该条件数的取值也反映了矩阵求逆和线性方程求解的精度。当 cond(X)或 cond(X, p)的值接近 1 时，表明该矩阵是良性的(更易于求逆)。

c = cond (X)返回 2 阶范数条件数，即矩阵 X 的最大奇异值与最小奇异值之比。

c = cond (X, p)返回矩阵 p 阶范数条件数，具体取值如表 6-1 所示。

表 6-1　　　　条件数返回值

p 的取值	cond (X, p) 的返回值
1	1 阶范数条件数
2	2 阶范数条件数
'fro'	Frobenius 范数条件数
Inf	无穷大范数条件数

举例：

(1) 对于任意一个 5×5 矩阵，表示为

```
A =
     1     1     1     1     1
     2     4     1     4     5
     3     6     1     0     7
     9     3     0     5     1
     4     7     1     0     3
```

在 MATLAB 命令窗口中输入：c = cond (A)，计算结果为：c = 21.5457。

在 MATLAB 命令窗口中输入：c = cond (A, 1)，计算结果为：c = 36.5983。

在 MATLAB 命令窗口中输入：c = cond (A, 2)，计算结果为：c = 21.5457。

在 MATLAB 命令窗口中输入：c = cond (A, inf)，计算结果为：c = 27.7799。

在 MATLAB 命令窗口中输入：c = cond (A, 'fro')，计算结果为：c = 27.0665。

(2) 若对于任意一个标量而言，例如 A = 8

在 MATLAB 命令窗口中输入：c = cond (A)，计算结果为：c = 1。

在 MATLAB 命令窗口中输入：c = cond (A, 1)，计算结果为：c = 1。

在 MATLAB 命令窗口中输入：c = cond (A, 2)，计算结果为：c = 1。

在 MATLAB 命令窗口中输入：c = cond (A, inf)，计算结果为：c = 1。

在 MATLAB 命令窗口中输入：c = cond (A, 'fro')，计算结果为：c = 1。

2. 特征值相关条件数

名称：condeig

和特征值相关的条件数。

语法：该函数有如下两种表达形式：

- c = condeig (A)
- [V, D, s] = condeig (A)

描述：c = cond (A)返回矩阵 A 的特征值的条件数向量。这些条件数是矩阵 A 的左右特征向量夹角余弦的倒数。

[V, D, s] = condeig (A)等效于[V, D] = eig (A)；s = condeig (A)。

条件数很大，表示矩阵 A 有可能有多重特征值。

举例：

对于任意一个 3×3 矩阵，表示为

```
A =
     15     32     71
     44     29     19
     52     18     34
```

在 MATLAB 命令窗口中输入：c = condeig (A)

计算结果为：

```
c =
     1.0116
     1.0436
     1.0508
```

在 MATLAB 命令窗口中输入：[V, D, s] = condeig (A)

计算结果为：

```
V =
    -0.6340    -0.8015     0.1510
    -0.5064     0.3769    -0.9027
    -0.5844     0.4642     0.4029
D =
   106.0092          0          0
          0   -41.1688          0
          0          0    13.1596
s =
```

1.0116
1.0436
1.0508

3. 行列式

名称：det

矩阵的行列式。

语法：d = det (X)

描述：d = det (X)返回方阵(行数和列数相同的矩阵)的行列式。若矩阵 X 中只包括整数，则 d 的值也为整数。

对于秩比较小的整数矩阵，可以用 det (X)=0 来进行矩阵奇异性的判断。有时也用“det (X) <允许值”来判断矩阵的奇异性，但由于允许值的选择有一定的难度，因此并不推荐使用该方法。另外，也可以利用函数 cond(X)进行矩阵奇异性或近似奇异性的检测。

MATLAB 内部进行矩阵行列式的计算时，首先采用高斯消去法对矩阵进行消元，得到一个上(或下)三角矩阵，然后将对角元素相乘就可以得到该矩阵的行列式。该过程可以用 MATLAB 函数(有关函数的意义可参见本书相应部分的说明)表示为：

```
[L, U] = lu(A)
s =  det(L)
det(A) = s*prod(diag(U))
```

举例

对于任意一个 3×3 矩阵，表示为

```
A =
     1     2     3
     4     5     6
     7     8     9
```

在 MATLAB 命令窗口中输入：d = det (A)

计算结果(即 A 矩阵的行列式)为：d = 0。这表示 A 矩阵为奇异矩阵。

若将 A 矩阵中的元素改为：A(3,3)=0。再在 MATLAB 命令窗口中输入：d = det (A)。则可得计算结果为 d = 27。

4. 范数

名称：norm

向量和矩阵的范数。

语法：该函数有如下两种表达形式：

- n = norm (A)
- n = norm (A, p)

描述：矩阵的范数是一个标量，用来表示和描述矩阵中各元素的量级。

norm 函数可以用来计算几种不同类型的矩阵范数：

n = norm (A)返回矩阵 A 的最大奇异值，即 max(svd(A))。

n = norm (A, p)返回矩阵 A 的依赖于 p 值的不同的范数，具体取值如表 6-2 所示。

表 6-2　　矩阵 A 对应于不同 p 值的范数

P 的取值	norm(X, p) 的返回值
1	1 阶范数，或矩阵各列元素和的最大值，即 max(sum(abs((A)))
2	最大奇异值，等价于 norm(A)
'fro'	矩阵 A 的 Frobenius 范数，即 sqrt(sum(diag(A'*A)))
Inf	无穷大范数，或矩阵各行元素和的最大值，即 max(sum(abs((A')))

当 A 为向量时，上述运算规则稍有不同

n = norm (A, p)，对任何满足条件 $1 \le p \le \infty$ 的 p，返回 sum(abs(A).^p)^(1/p) 的值。

n = norm (A)返回 norm (A, 2)的值。

n = norm (A, inf)返回 max(abs(A))的值。

n = norm (A, -inf) 返回 min(abs(A))的值。

另外，利用 norm(A)/sqrt(n)可以得到矩阵 A 中各元素的方根平均值。

举例：

(1) 对于任意一个 3×3 矩阵，表示为

A =

```
    1    2    3
    4    5    6
    7    8    9
```

在 MATLAB 命令窗口中输入：n = norm (A)，

计算结果(即 A 矩阵的范数)为：n = 16.8481。

在 MATLAB 命令窗口中输入：n = norm (A,1)，

计算结果(即 A 矩阵的 1 阶范数)为：n = 18。

在 MATLAB 命令窗口中输入：n = norm (A,2)，

计算结果(即 A 矩阵的最大奇异值)为：n = 16.8481。

在 MATLAB 命令窗口中输入：n = norm (A, inf)，

计算结果(即 A 矩阵的无穷大范数)为：n = 24。

在 MATLAB 命令窗口中输入：n = norm (A, 'fro')，

计算结果(即 A 矩阵的 Frobenius 范数)为：n = 16.8819。

(2) 对于任一向量 A=[3　4　5　6]

在 MATLAB 命令窗口中输入：n = norm (A)，

计算结果为：n = 9.2736。

在 MATLAB 命令窗口中输入：n = norm (A,1)，

计算结果为：n = 18。

在 MATLAB 命令窗口中输入：n = norm (A,2)，

计算结果为：n = 9.2736。

在 MATLAB 命令窗口中输入：n = norm (A, inf)，

计算结果为：n = 6。

在 MATLAB 命令窗口中输入：n = norm (A, -inf)，

计算结果为：n = 3。

5. 0 空间

名称：null

矩阵的 0 空间。

语法：B = null (A)

描述：B = null (A)返回矩阵 A 的 0 空间的标准化正交基。同时满足：B'*B = I，A*B 中含有可以忽略的元素。

举例：

对于任意一个 3×3 矩阵，表示为

```
A =
      1     4     5
      8     2     9
      1     0     4
```

在 MATLAB 命令窗口中输入：B = null (A)，

计算结果为：B = Empty matrix: 3-by-0。

6. 正交化空间

名称：orth

矩阵的正交化空间。

语法：B = orh (A)

描述：B = orth (A)返回矩阵 A 的标准化正交基。矩阵 B 的列和矩阵 A 的列位于同一空间，而且矩阵 B 的各列之间彼此是正交的，因此 B'*B = eye(rank(A))。矩阵 B 的列数等于矩阵 A 的秩。

举例：

对于任意一个 3×3 矩阵，表示为

```
A =
      1     4     5
      8     2     9
      1     0     4
```

在 MATLAB 命令窗口中输入：B = orth (A)，

计算结果为：

```
B =
    0.3986    0.8978    0.1873
    0.8762   -0.4331    0.2112
    0.2707    0.0800   -0.9593
```

7. 矩阵的秩

名称：rank

求解矩阵的秩。

语法：该函数有如下两种表达式：

- k = rank (A)
- k = rank (A, tol)

描述：rank 函数可以用来确定矩阵中线性无关的行或列的数目。

k = rank (A) 返回矩阵 A 中大于给定允许值的奇异值的数目，即：max(size(A))*norm(A)*eps。

k = rank (A, tol)返回矩阵 A 中大于 tol 值的奇异值的数目。

在数学上有许多种方法可以用来计算矩阵的秩。在 MATLAB 中，常常采用基于奇异值分解的方法，或者称为 SVD 法。相对其他方法而言，该方法是最耗时的，但计算结果往往是最可靠的。

SVD 法用 MATLAB 函数可以描述为：

```
s = svd(A);
tol = max(size(A))*s(1)*eps;
r = sum(s > tol);
```

举例：

对于任意一个 3×3 矩阵，表示为

```
A =
      1     4     5
      8     2     9
      1     0     4
```

在 MATLAB 命令窗口中输入：k = rank (A)，

计算结果(矩阵 A 的秩)为：k = 3。

在 MATLAB 命令窗口中输入：k = rank (A, 5)，

计算结果为：k = 1。

8. 矩阵的逆条件数

名称：rcond

求解矩阵的逆条件数。

语法：c = rcond (A)

描述：c = rcond (A)返回使用 LINPACK 条件估计因子计算得到的矩阵 A 的 1 阶范数条件数的倒数值。如果矩阵 A 是一个良态矩阵，则 rcond (A)的返回值接近于 1.0；若 A 是一个病态矩阵，rcond (A)的返回值接近于 0.0。

与 cond 函数相比，rcond 函数在计算上更有效，但计算结果的可靠性不好。

举例：

(1) 对于任意一个 3×3 矩阵，表示为

```
A =
      1     4     5
      8     2     9
      1     0     4
```

在 MATLAB 命令窗口中输入：c = rcond (A)，

计算结果为：c = 0.0773。表明该矩阵接近于病态。

(2) 若另取一个 3×3 矩阵：

```
A =
     1     0     0
     0     1     0
     0     0     1
```

在 MATLAB 命令窗口中输入：c = rcond (A)，

计算结果为：c = 1。表明该矩阵是良态矩阵。

9. 压缩的行阶梯形

名称：rref, rrefmovie

求解压缩的行阶梯形。

语法：该函数有如下几种表达形式：

- R = rref (A)
- [R, jb] = rref (A)
- [R, jb] = rref (A, tol)
- rrefmovie (A)

描述：R = rref (A)返回采用高斯—约当消去法求解得到的 A 矩阵的压缩的行阶梯形。

[R, jb] = rref (A)还返回满足如下条件的向量 jb：

1，length(jb)在数值上等于矩阵 A 的秩。

2，x(jb)为线性方程组 Ax = b 的限制变量。

3，A(:, jb)为矩阵 A 所在空间的基。

4，R(1:r, jb)是 r×r 阶单位矩阵。

[R, jb] = rref(A, tol)返回基于给定允许值 tol 的 A 矩阵的压缩的行阶梯形和向量 jb。

rrefmovie (A)用来显示 MATLAB 的整个求解过程。

举例：

对于 4 阶魔方矩阵 A = magic (4)：

```
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

在 MATLAB 命令窗口中输入：R = rref (A)，

计算结果为：

```
R =
     1     0     0     1
     0     1     0     3
     0     0     1    -3
     0     0     0     0
```

在 MATLAB 命令窗口中输入：[R, jb] = rref (A)，

计算结果为：

```
R =
     1     0     0     1
     0     1     0     3
     0     0     1    -3
     0     0     0     0
jb =
     1     2     3
```

在 MATLAB 命令窗口中输入：[R, jb] = rref (A, 3)，

计算结果为：

```
R =
       1          0         1/18         0
       0          1        19/18         0
       0          0          0           1
       0          0          0           0
jb =
     1     2     4
```

在 MATLAB 命令窗口中输入：rrefmovie (A)，

此时在屏幕上会依次显示出：

```
Original matrix:
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
Press any key to continue…
pivot = A (1, 1)
A =
       1          1/8         3/16        13/16
       5          11          10            8
       9           7           6           12
       4          14          15            1
Press any key to continue…
eliminate in column 1
A =
       1          1/8         3/16        13/16
       5          11          10            8
       9           7           6           12
       4          14          15            1
Press any key to continue. . .
```

```
A =
        1          1/8         3/16        13/16
        0         83/8       145/16        63/16
        0         47/8        69/16        75/16
        0         27/2         57/4         -9/4
Press any key to continue. . .
swap rows 2 and 4
A =
        1          1/8         3/16        13/16
        0         27/2         57/4         -9/4
        0         47/8        69/16        75/16
        0         83/8       145/16        63/16
Press any key to continue. . .
pivot = A(2,2)
A =
        1          1/8         3/16        13/16
        0            1        19/18         -1/6
        0         47/8        69/16        75/16
        0         83/8       145/16        63/16
Press any key to continue. . .
eliminate in column 2
A =
        1          1/8         3/16        13/16
        0            1        19/18         -1/6
        0         47/8        69/16        75/16
        0         83/8       145/16        63/16
Press any key to continue. . .
A =
        1            0         1/18          5/6
        0            1        19/18         -1/6
        0            0        -17/9         17/3
        0            0        -17/9         17/3
Press any key to continue. . .
pivot = A(3,3)
A =
        1            0         1/18          5/6
        0            1        19/18         -1/6
        0            0            1           -3
        0            0        -17/9         17/3
```

```
Press any key to continue...
eliminate in column 3
A =
     1          0          1/18        5/6
     0          1          19/18       -1/6
     0          0          1           -3
     0          0          -17/9       17/3
Press any key to continue...
A =
     1          0          0           1
     0          1          0           3
     0          0          1           -3
     0          0          0           *
Press any key to continue...
column 4 is negligible
A =
     1          0          0           1
     0          1          0           3
     0          0          1           -3
     0          0          0           0
```

1. 子空间的角度

名称：subspace

两个子空间的角度。

语法：theta = subspace(A,B)

描述：theta = subspace(A,B)用来求解由矩阵 A 和矩阵 B 的列向量所组成的两个子空间的夹角(以弧度为单位)。若 A 和 B 均为单位长度的向量，则函数 subspace(A,B)计算得到的值与函数 acos(A'*B)计算得到的值相同。

若两个子空间之间的夹角非常小，则表明这两个子空间是线性相关的。

举例：

考虑由 Hadamard 矩阵(列向量相互正交)派生出的两个子空间，在 MATLAB 窗口中输入：

H = hadamard(8)，得到：

```
H =
     1     1     1     1     1     1     1     1
     1    -1     1    -1     1    -1     1    -1
     1     1    -1    -1     1     1    -1    -1
     1    -1    -1     1     1    -1    -1     1
     1     1     1     1    -1    -1    -1    -1
     1    -1     1    -1    -1     1    -1     1
     1     1    -1    -1    -1    -1     1     1
```

```
     1    -1    -1     1    -1     1     1    -1
```

输入：A = H (:,2:4)，得到：

```
A =
     1     1     1
    -1     1    -1
     1    -1    -1
    -1    -1     1
     1     1     1
    -1     1    -1
     1    -1    -1
    -1    -1     1
```

输入：B = H (:,5:8)，得到：

```
B =
     1     1     1     1
     1    -1     1    -1
     1     1    -1    -1
     1    -1    -1     1
    -1    -1    -1    -1
    -1     1    -1     1
    -1    -1     1     1
    -1     1     1    -1
```

需要指出的是，在应用 subspace 函数时，矩阵 A 和矩阵 B 的维数可以不相同。例如本例中矩阵 A 有 3 列，而矩阵 B 有 4 列。从几何上来讲，所谓两个子空间的夹角，实际上就是镶嵌于一个多维空间的两个超平面之间的夹角。

在 MATLAB 命令窗口中输入：theta = subspace(A,B)，

得到的结果为：theta = 1.5708。

当 A 与 B 正交时，得到的 theta 值为$\pi/2$ (即 90 度)。

在 MATLAB 命令窗口中输入：theta - pi/2，

得到的结果为：ans = 0。

11. 矩阵的迹

名称：trace

求解矩阵的迹。

语法：b = trace (A)

描述：b = trace (A)返回的是矩阵 A 的迹(也即，矩阵 A 的主对角元素的和)。

举例：

对于任意一个 3×3 矩阵：

```
A =
     1     4     5
     8     2     9
```

```
    1     0     4
```

在 MATLAB 命令窗口中输入：b = trace (A)，

计算结果为：b = 7，即该矩阵的迹为 7。

6.2　线性方程

1. Cholesky 分解

名称：chol

Cholesky 分解。

语法：该函数有如下两种表达形式：

- R = chol (X)
- [R, p] = chol (X)

描述：MATLAB 解线性方程基于以下三种分解：

1，正定系数矩阵的 Cholesky 分解；

2，系数矩阵为普通方阵的高斯消去法；

3，长方阵的正交分解。

上述三种分解分别用 chol、lu 和 qr 函数来完成。本处先介绍 chol 函数，另外两个函数将在后面的内容中专门介绍。

Cholesky 分解把矩阵分解为上三角矩阵和其转置的乘积。数学表示为：X = R'*R，其中 R 为上三角矩阵。如果 X 不是正定矩阵，将会出现出错信息提示。

如果复数矩阵满足 Hermite 正定，则也有 Cholesky 分解。

使用函数[R, p] = chol (X)时，不会出现出错信息提示。若 X 正定，则 p=0，R 和上面的相同；若 X 不正定，则 p 为正整数，R 为 q (=p-1)阶上三角矩阵，并满足：R'*R = X(1:q,1:q)。

举例：

下面求 6 阶 pascal 矩阵的 Cholesky 分解：

```
X = pascal (6)
X =
     1     1     1     1     1     1
     1     2     3     4     5     6
     1     3     6    10    15    21
     1     4    10    20    35    56
     1     5    15    35    70   126
     1     6    21    56   126   252
```

在 MATLAB 命令窗口中输入：R = chol (X)，

计算结果为：

```
R =
     1     1     1     1     1     1
     0     1     2     3     4     5
```

```
        0    0    1    3    6   10
        0    0    0    1    4   10
        0    0    0    0    1    5
        0    0    0    0    0    1
```

若在 MATLAB 命令窗口中输入：[R, p] = chol (X)，

输出的结果为：

```
R =
        1    1    1    1    1    1
        0    1    2    3    4    5
        0    0    1    3    6   10
        0    0    0    1    4   10
        0    0    0    0    1    5
        0    0    0    0    0    1
p = 0
```

可以看出，由于矩阵 X 正定，所以 R = chol (X)和[R, p] = chol (X)得到的 R 均相同，并且 p=0。

而对于不正定的矩阵，例如：

```
X =
        1    1    1    1    3    1
        0    2    4    2    3    1
        1    2    3    4    5    1
        5    6    7    8    7    1
        1    6    2    3    4    1
        2    3    4    1    2    3
```

当在 MATLAB 命令窗口中输入 R = chol (X)时，会出现如下出错信息：

??? Error using ==> chol   Matrix must be positive definite.

当在 MATLAB 命令窗口中输入[R, p] = chol (X)时，计算结果为：

```
R =
        1    1
        0    1
p = 3
```

2. 矩阵求逆

名称：inv

矩阵求逆。

语法：Y = inv (X)

描述：Y = inv (X)返回方阵 X 的逆。若 X 矩阵奇异或近似奇异，将会出现错误信息。

在实际使用中，只进行单纯地求解矩阵逆的运算的情况很少。更多的是用在线性方程组的求解过程中。

例如在求解线性方程组 Ax = b 时，可以采用公式 x = inv(A)*b 来进行。另一种方法则

是采用前面介绍的矩阵求解符‘\’进行，即 x = A\b。由于后者在求解过程中只是用到了高斯消去法而不需要进行矩阵的求逆运算，因此数值计算结果更准确，占用内存更小，算的更快。因此在求解线性方程组时，建议使用矩阵求解符‘\’进行。

举例：

对于任意一个 3×3 矩阵：

```
X =
      1     4     5
      8     2     9
      1     3     4
```

在 MATLAB 命令窗口中输入：R = inv (X)，

计算结果(即 X 矩阵的逆)为：

```
R =
   19.0000    1.0000  -26.0000
   23.0000    1.0000  -31.0000
  -22.0000   -1.0000   30.0000
```

3. 最小二乘解

名称：lscov

已知协方差的最小二乘解。

语法：该函数有如下两种表达形式：

- x = lscov(A,b,V)
- [x,dx] = lscov(A,b,V)

描述：x = lscov(A,b,V)返回求解方程 A*x = b + e (其中 e 随协方差 V 呈正态分布)时得到的向量 x。其中矩阵 A 必须是 m×n 阶的，并且 m>n。

[x,dx] = lscov(A,b,V)返回 x 在 dx 变化时的标准误差。求解系数的标准误差在统计学中可以用如下公式来表示(有关函数的意义可参见本书中的说明)：

```
mse = B'*(inv(V)-inv(V)*A*inv(A'*inv(V)*A)*A'*inv(V))*B./(m-n)
dx = sqrt(diag(inv(A'*inv(V)*A)*mse))
```

所谓最小二乘解，实际上就是求使得(A*x-b)'*inv(V)*(A*x-b)最小的 x 的值。采用经典线性代数对这一问题进行求解时，其结果为 x = inv(A'*inv(V)*A)*A'*inv(V)*b。但采用 lscov 函数求解时，首先需要对 A 矩阵进行 QR 分解，然后用 V 修正 Q。

举例：

对于给定的 4×3 矩阵：

```
A =
      1     4     5
      8     2     9
      1     3     4
      2     6     8
```

和列向量：

b =

```
    3
    5
    7
    9
```

以及协方差：

```
V =
    3    3    3    2
    4    7    1    6
    9    2    6    1
    1    0    3    9
```

在 MATLAB 命令窗口中输入：x = lscov(A,b,V)，

计算结果为：

```
x =
  -77.2000
  -92.4000
   89.8000
```

4. LU 分解

名称：lu

矩阵的 LU 分解。

语法：该函数有如下几种表达形式：

- [L, U] = lu (X)
- [L, U, P] = lu (X)
- lu (X)

描述：前面在介绍 chol 函数时已经讲到，MATLAB 解线性方程基于三种分解，而且分别对应于三个函数 chol、lu 和 qr。其中，第二种分解方法叫做 LU 分解，又称 Gauss 消去法。该方法可以把任意方阵分解成下三角矩阵的基本变换形式(行交换)和上三角矩阵的乘积。其数学表达式为：X=L*U。其中 L 为下三角矩阵的基本变换形式，U 为上三角矩阵。

[L, U] = lu (X)返回一个上三角矩阵 U 和一个下三角矩阵的基本变换形式 L (一个下三角矩阵和置换矩阵的乘积)，并满足 X=L*U。

[L, U, P] = lu (X)返回一个上三角矩阵 U、一个下三角矩阵 L 和置换矩阵 P，并满足 L*U=P*X。

lu (X)返回 LINPACK 程序 ZGEFA 的输出值。

举例：

对于给定的 3×3 矩阵：

```
X =
    1    2    3
    4    5    6
    7    8    0
```

首先在 MATLAB 命令窗口中输入：[L, U] = lu (X)，

MATLAB 的分解结果为：

```
L =
    0.1429    1.0000         0
    0.5714    0.5000    1.0000
    1.0000         0         0
U =
    7.0000    8.0000         0
         0    0.8571    3.0000
         0         0    4.5000
```

从上面的结果可以看出，L 是一个下三角矩阵的基本变换形式，U 是一个上三角矩阵。为了验证分解结果是否正确，可进行如下计算：

在 MATLAB 命令窗口中输入：L*U，可得到输出结果：

```
ans =
     1     2     3
     4     5     6
     7     8     0
```

即与原始矩阵 X 完全相同。

为了得到置换矩阵，在 MATLAB 命令窗口中输入：[L, U, P] = lu (X)。此时会得到与上面相同的上三角矩阵 U，但 L 矩阵则进行了重新排序，而且得到一个置换矩阵，如下所示：

```
L =
        1.0000         0         0
        0.1429    1.0000         0
        0.5714    0.5000    1.0000
U =
        7.0000    8.0000         0
             0    0.8571    3.0000
             0         0    4.5000
P =
        0     0     1
        1     0     0
        0     1     0
```

为了证实此时 L*U 是原始矩阵 X 的基本变换形式，可进行如下操作：

在 MATLAB 命令窗口中输入 L*U，并从 P*X 中减掉 L*U，即输入 P*A - L*U。最后可得：

```
ans =
     0     0     0
     0     0     0
     0     0     0
```

这证明了 L*U 确实是原始矩阵 X 的基本变换形式。

矩阵的 LU 分解使得 MATLAB 解系数矩阵为方阵的方程时，把方程 A*x=b 变为 x=U\(L\b)，这样计算速度将会大大提高。事实上，在 MATLAB 里，矩阵的求逆和求行列式都是通过 LU 分解来实现的：det(A)=det(L)*det(U)，inv(A)=inv(U)*inv(L)。

5. 非负约束的线性最小二乘

名称：lsqnonneg

非负约束的线性最小二乘。

语法：该函数有如下几种表达形式：

- x = lsqnonneg(C,d)
- x = lsqnonneg(C,d,x0)
- x = lsqnonneg(C,d,x0,options)
- [x,resnorm] = lsqnonneg(...)
- [x,resnorm,residual] = lsqnonneg(...)
- [x,resnorm,residual,exitflag] = lsqnonneg(...)
- [x,resnorm,residual,exitflag,output] = lsqnonneg(...)
- [x,resnorm,residual,exitflag,output,lambda] = lsqnonneg(...)

描述：x = lsqnonneg(C,d)返回在 x≥0，且 C 和 d 均为实数的条件下，使 norm(C*x-d)取最小值的向量 x。

x = lsqnonneg(C,d,x0)当 $x_0 \geq 0$ 时，把 x_0 作为起始点，否则使用缺省值(缺省值为原点)。

x = lsqnonneg(C,d,x0,options)最小化结构 options 中指定的优化参数。用户可以使用 optimset 函数来定义这些参数。lsqnonneg 函数中所使用的 options 结构的参数包括：

1，Display — 表示显示的级别。

2，Off — 不会显示输出结果。

3，Iter — 在每一个迭代步显示输出结果。

4，Final — 仅显示最终输出结果。

5，TolX — 迭代求解 x 的允许值。

[x,resnorm] = lsqnonneg(...)返回残差 2 阶范数的平方，即 norm(C*x-d)^2。

[x,resnorm,residual] = lsqnonneg(...)返回残差 C*x-d。

[x,resnorm,residual,exitflag] = lsqnonneg(...)返回一个 exitflag 值，该值用来描述 lsqnonneg 函数的退出条件：

1，>0 表示函数收敛于解 x。

2，=0 表示迭代次数已经超过允许值。此时，增加收敛允许值(Tol x)会函数收敛于解 x。

3，<0 表示函数不收敛于解 x。

[x,resnorm,residual,exitflag,output] = lsqnonneg(...)返回一个 output 结构，该结构中包含有关操作信息：

1，output.iterations — 已经迭代数。

2，output.algorithm — 使用的运算法则。

[x,resnorm,residual,exitflag,output,lambda] = lsqnonneg(...)返回一个对偶向量 lambda。并且，当 x (i)等于(或近似等于)0 时，lambda(i)≤0；当 x (i)>0 时，lambda(i)等于(或近似等于)0。

举例：

对于如下给定的矩阵和向量：

```
C =
    0.0372    0.2869
    0.6861    0.7071
    0.6233    0.6245
    0.6344    0.6170
d =
    0.8587
    0.1781
    0.0747
    0.8405
```

经过计算可以得到：

```
[C\d lsqnonneg(C,d)] =
        -2.5627         0
         3.1108    0.6929
[norm(C*(C\d)-d) norm(C*lsqnonneg(C,d)-d)] = 0.6674    0.9118
```

6. Moore-Penrose 伪逆

名称：pinv

矩阵的 Moore-Penrose 伪逆。

语法：该函数有如下两种表达形式：

- B = pinv(A)
- B = pinv(A,tol)

描述：前面已经讲到，如果矩阵 A 为方阵且非奇异，则方程 A*X=I 和 X*A=I 的解称为矩阵 A 的逆，并且用 A^{-1} 表示。矩阵的逆可用函数 inv 来进行求解。

当矩阵为长方形阵时，方程 A*X=I 和 X*A=I 中至少有一个无解。因此就引入了伪逆的概念以便在某种程度上代表矩阵的逆。具体来讲，所谓矩阵 A 的伪逆就是指满足如下四个条件的矩阵 B (它和 A 矩阵的转置有相同的维数)：

1，A*B*A = A，

2，B*A*B = B，

3，A*B 是 Hermitian 矩阵，

4，B*A 是 Hermitian 矩阵。

计算是基于函数 svd(A)进行的，任何小于 1 的奇异值都将被作为 0 来进行处理。

B = pinv(A)返回矩阵 A 的 Moore-Penrose 伪逆。

B = pinv(A,tol)返回矩阵 A 的 Moore-Penrose 伪逆，并重载缺省的允许值。

举例：

如果矩阵 A 是一个方阵而且不奇异，那么用 pinv (A)来计算 A 的逆，在计算用时上是不合算的。如果 A 不是一个方阵，或者是一个方阵但奇异，那么 inv(A)就不能运行。在这种情况下，就只可以使用函数 pinv(A)进行矩阵求逆运算。而且函数 pinv(A)具有函数 inv(A)的部分(而不是全部) 性质。

按如下步骤得到一个 8×6 矩阵：A = magic(8)；A = A(:,1:6)，即：

```
A =
    64     2     3    61    60     6
     9    55    54    12    13    51
    17    47    46    20    21    43
    40    26    27    37    36    30
    32    34    35    29    28    38
    41    23    22    44    45    19
    49    15    14    52    53    11
     8    58    59     5     4    62
```

在 MATLAB 命令窗口中输入：B = pinv(A)，可得：

```
B =
Columns 1 through 8
   0.0177   -0.0165   -0.0164    0.0174    0.0173   -0.0161   -0.0160    0.0170
  -0.0121    0.0132    0.0130   -0.0114   -0.0112    0.0124    0.0122   -0.0106
  -0.0055    0.0064    0.0060   -0.0043   -0.0040    0.0049    0.0045   -0.0028
  -0.0020    0.0039    0.0046   -0.0038   -0.0044    0.0064    0.0070   -0.0063
  -0.0086    0.0108    0.0115   -0.0109   -0.0117    0.0139    0.0147   -0.0141
   0.0142   -0.0140   -0.0149    0.0169    0.0178   -0.0176   -0.0185    0.0205
```

若在 MATLAB 命令窗口中输入：B = inv(A)，则会出现如下错误信息：

```
??? Error using ==> inv    Matrix must be square.
```

在 MATLAB 命令窗口中输入：tol = 1.0000e-015；B = pinv(A,tol)，则可得输出结果为：

```
B = 1.0e+014 *
  Columns 1 through 8
  -0.3538   -0.2054    0.0840    0.1526    0.4172    0.0766    0.0448   -0.2160
  -0.0329   -0.2536   -0.2326    0.1158   -0.8291   -0.0555    0.5417    0.7461
  -0.2819    0.1241    0.2967   -0.0971    1.5315    0.4259   -0.8467   -1.1526
   0.4380    0.1792   -0.3762   -0.4384   -0.3906    0.7491   -0.5522    0.3910
  -0.0842    0.0261    0.2922    0.2858   -0.0266   -0.8257    0.5074   -0.1750
   0.3147    0.1295   -0.0641   -0.0187   -0.7024   -0.3704    0.3049    0.4064
```

如果矩阵 A 的行数多于列数而且不是满秩的，那么最小二乘问题 minimize norm(A*x-b)就没有唯一确定的解(可能有多个解)。其中的两个解为：x = pinv(A)*b 和 y = A\b。这两个解与其他解的不同之处在于：norm(x)小于任何其他解的范数；y 有最少的非零元素。

例如采用 b = 260*ones(8,1)得到向量 b，可得：

```
b =
   260
   260
   260
   260
   260
```

```
    260
    260
    260
```

在 MATLAB 命令窗口中输入：x = pinv(A)*b，可求得其中的一个解为：

```
x =
      1.1538
      1.4615
      1.3846
      1.3846
      1.4615
      1.1538
```

在 MATLAB 命令窗口中输入：y = A\b，可求得其中的另一个解为：

```
y =
      4.0000
      5.0000
           0
           0
           0
     -1.0000
```

7. 正交三角分解

名称：qr

矩阵的正交三角分解。

语法：该函数有如下几种表达形式：

- [Q, R] = qr(X)
- [Q, R, E] = qr(X)
- [Q, R] = qr(X,0)
- [Q, R, E] = qr(X,0)
- A = qr(X)

描述：前面在介绍 chol 函数时已经讲到，MATLAB 解线性方程基于三种分解，而且分别对应于三个函数 chol、lu 和 qr。其中，第三种分解方法叫做 QR 分解，又称正交三角分解。在介绍 QR 分解之前，先介绍正交矩阵的概念，所谓正交矩阵就是指满足 Q'*Q=I 的实矩阵 Q。

QR 分解就是把任何的长方阵分解为正交矩阵和上三角矩阵的初等变换形式的乘积。

[Q, R] = qr(X)返回一个和矩阵 X 同维数的上三角矩阵 R，和一个正交矩阵 Q，并满足 X=Q*R。

[Q, R, E] = qr(X)返回一个置换矩阵 E，一个上三角矩阵 R 和一个正交矩阵，并且满足 X*E= Q*R。

[Q, R] = qr(X,0)和[Q, R, E] = qr(X,0)表示进行紧凑形分解。其中 E 是一个置换向量。

A = qr(X)返回 LINPACK 子程序 ZQRDC 的输出结果。

举例：

对于矩阵

```
A =
     1     2     3
     4     5     6
     7     8     9
    10    11    12
```

可以看出，该矩阵不是一个满秩矩阵，因为中间一列等于其他两列的平均值。而且从下面的分解结果也可以看出。

在 MATLAB 命令窗口中输入：[Q, R] = qr(A)，可得分解结果为：

```
Q =
   -0.0776   -0.8331   -0.2636   -0.4801
   -0.3105   -0.4512    0.7093    0.4437
   -0.5433   -0.0694   -0.6278    0.5530
   -0.7762    0.3124    0.1821   -0.5166
R =
  -12.8841  -14.5916  -16.2992
         0   -1.0413   -2.0826
         0         0    0.0000
         0         0         0
```

R 矩阵的第 3 行全部为 0，这表明矩阵 R(从而矩阵 A)不是满秩的。

在 MATLAB 命令窗口中输入：[Q, R, E] = qr(A)，可得分解结果为：

```
Q =
   -0.1826   -0.8165    0.5251   -0.1557
   -0.3651   -0.4082   -0.5842    0.5990
   -0.5477         0   -0.4071   -0.7309
   -0.7303    0.4082    0.4661    0.2876
R =
  -16.4317  -12.7802  -14.6059
         0    1.6330    0.8165
         0         0   -0.0000
         0         0         0
E =
     0     1     0
     0     0     1
     1     0     0
```

其中 E 为置换矩阵。

在 MATLAB 命令窗口中输入：[Q, R] = qr(A,0)，可得分解结果为：

```
Q =
```

```
   -0.0776   -0.8331   -0.2636
   -0.3105   -0.4512    0.7093
   -0.5433   -0.0694   -0.6278
   -0.7762    0.3124    0.1821
R =
  -12.8841  -14.5916  -16.2992
         0   -1.0413   -2.0826
         0         0   -0.0000
```

在 MATLAB 命令窗口中输入：[Q, R, E] = qr(A,0)，可得分解结果为：

```
Q =
   -0.1826   -0.8165    0.5251
   -0.3651   -0.4082   -0.5842
   -0.5477         0   -0.4071
   -0.7303    0.4082    0.4661
R =
  -16.4317  -12.7802  -14.6059
         0    1.6330    0.8165
         0         0   -0.0000
E =
     3     1     2
```

在 MATLAB 命令窗口中输入：B = qr(A)，可得：

```
B =
  -12.8841  -14.5916  -16.2992
    0.3105   -1.0413   -2.0826
    0.5433   -0.3506   -0.0000
    0.7762   -0.9124   -0.9635
```

6.3 特征值和奇异值

1. 提高特征值精度

名称：balance

提高特征值精度的选项。

语法：该函数有如下两种表达形式：

- [D,B] = balance(A)
- B = balance(A)

描述：非对称矩阵通常含有性态比较差的特征值。因此在这类矩阵中的一个小的扰动，会导致特征值的巨大变化。联系矩阵扰动大小和特征值扰动大小二者之间关系的量是特征向量矩阵的条件数，即 cond(V) = norm(V)*norm(inv(V))，其中[V,D] = eig(A)。需要指出的是，

矩阵 A 本身的条件数与特征值问题没有关系。

此处介绍的平衡函数 balance 就是试图将特征向量的所有病态条件浓缩到一个对角矩阵中。所谓平衡处理，并不是要把非对称矩阵变换成一个对称矩阵，而只是尽量作到矩阵中每一行的范数等于相应列的范数。而且对角矩阵的元素被限制为 2 的整数次幂可以避免引入计算误差。

[D,B] = balance(A)返回一个对角矩阵 D，它所包含的元素是 2 的整数次幂，和一个平衡矩阵，并且满足：B = D\A*D。若 A 是对称矩阵，则 B = A，而且 D 是一个单位矩阵。

B = balance(A)返回平衡矩阵。

MATLAB 的特征值函数为 eig(A)。在计算 A 的特征值之前，MATLAB 会自动进行平衡处理。若关掉平衡处理功能，需加入 nobalance 选项，即 eig(A,'nobalance')。

另外，由于平衡处理有可能会破坏某些矩阵的特性，因此在使用时要进行慎重考虑。

举例：

矩阵 A 的右上方为很大的数，而左下方为很小的数，而且并不对称：

```
A =
  1.0e+004 *
    0.0001    0.0100    1.0000
    0.0000    0.0001    0.0100
    0.0000    0.0000    0.0001
```

在 MATLAB 命令窗口中输入：[D,B] = balance(A)，可得到一个对角矩阵 D，和一个在形式上比矩阵 A 更趋进于对称的平衡矩阵 B：

```
D =
  1.0e+003 *
    2.0480         0         0
         0    0.0320         0
         0         0    0.0003
B =
    1.0000    1.5625    1.2207
    0.6400    1.0000    0.7813
    0.8192    1.2800    1.0000
```

为了验证平衡函数对特征向量精度的影响，首先计算矩阵 A 的特征向量。在 MATLAB 命令窗口中输入：[V,E] = eig(A) (该函数的意义请参见相应的函数说明)，可得：

```
V =
   -1.0000    0.9999   -1.0000
    0.0050    0.0100    0.0034
    0.0000    0.0001    0.0001
E =
     0     0     0
     0     3     0
     0     0     0
```

从矩阵 V 的结果中可以看出，所有三个列向量的第一个元素的值都远远大于其他元素的值。这表明矩阵 V 是病态的，而且经计算可知，cond(V)= 1.7484e+05。

下面比较矩阵 B 的情况，在 MATLAB 命令窗口中输入：[V,E] = eig(B)，可得：

```
V =
   -0.8873    0.6933   -0.8642
    0.2839    0.4437    0.1887
    0.3634    0.5679    0.4664
E =
     0     0     0
     0     3     0
     0     0     0
```

可以看出，此时求得的特征向量表现出良好的性态，而且 cond(V)= 14.9035。矩阵的病态被浓缩到对角矩阵 D 中，而且 cond(D)=8192。

2. 复数对角型转换为实数对角型

名称：cdf2rdf

复数对角型转换为实数对角型。

语法：[V, D] = cdf2rdf(V, D)

描述：[V, D] = cdf2rdf(V, D)将复数矩阵转换为实数矩阵。

举例：

对如下所示的矩阵，

```
X =
     1     2     3
     0     4     5
     0    -5     4
```

利用函数[V,D] = eig(X)求解其特征值，可得一对复特征值：

```
V =
   1.0000             0.4002 - 0.0191i   0.4002 + 0.0191i
        0             0.6479             0.6479
        0                  0 + 0.6479i        0 - 0.6479i
D =
   1.0000                  0                  0
        0             4.0000 + 5.0000i        0
        0                  0             4.0000 - 5.0000i
```

利用函数[V, D] = cdf2rdf(V, D)将复矩阵转换为实数矩阵，得到：

```
V =
    1.0000    0.4002   -0.0191
         0    0.6479         0
         0         0    0.6479
D =
```

```
    1     0     0
    0     4     5
    0    -5     4
```

3. 求特征值和特征向量

名称：eig

求特征值和特征向量。

语法：该函数有如下几种表达形式：

- d = eig(A)
- [V, D] = eig(A)
- [V, D] = eig(A, 'nobalance')
- d = eig(A,B)
- [V, D] = eig(A, B)

描述：A 为方阵，且它的线性独立的特征向量个数等于矩阵 A 的阶数。

d = eig(A)返回矩阵 A 的特征值向量。

[V, D] = eig(A)返回特征值对角矩阵 D 和模态矩阵 V。V 矩阵的列向量就是 A 矩阵的全部右特征向量，并且满足 A*V = V*D。

[V, D] = eig(A, 'nobalance')表示在求解特征值和特征向量之前，不进行平衡处理。如前所述，通常情况下平衡处理可以提高输入矩阵的性态，并能够提高特征值和特征向量的计算精度。但由于平衡处理有可能会破坏某些矩阵的特性，因此在使用时要进行慎重考虑。在需要时，可以使用 eig 函数的 nobalance 选项取消求解前的平衡处理。

当 A 和 B 均为方阵时，d = eig(A,B)函数返回一个包含广义特征值的向量。

[V, D] = eig(A, B)返回一个广义特征值对角矩阵 D 和一个包含相应广义特征向量的模态矩阵 V，并且满足：A*V = B*V*D。V 矩阵中的特征向量均进行了缩放，使得每个向量的范数均为 1。

举例：

对于如下所示的矩阵：

```
B =
     3.0000   -2.0000   -0.9000    0.0000
    -2.0000    4.0000   -1.0000   -0.0000
    -0.0000    0.0000   -1.0000         0
    -0.5000   -0.5000    0.1000    1.0000
```

在 MATLAB 窗口中输入：d = eig(B)，可求得矩阵的特征值向量为：

```
d =
     5.5616
     1.4384
     1.0000
    -1.0000
```

在 MATLAB 窗口中输入：[V, D] = eig(B)，可以求得矩阵 B 的特征值对角矩阵 D 和模态矩阵 V：

```
V =
   -0.6153    0.4176   -0.0000   -0.3187
    0.7881    0.3261   -0.0000   -0.2843
    0.0000    0.0000    0.0000   -0.7844
   -0.0189   -0.8481    1.0000    0.4498
D =
    5.5616         0         0         0
         0    1.4384         0         0
         0         0    1.0000         0
         0         0         0   -1.0000
```

为了验证结果是否正确，在 MATLAB 窗口中输入：B*V -V*D，可得：

```
ans =
         0   -0.0000   -0.0000    0.0000
    0.0000   -0.0000    0.0000    0.0000
    0.0000    0.0000   -0.0000    0.0000
    0.0000    0.0000    0.0000    1.1227
```

表明所求得的特征值和特征向量是正确的。

在 MATLAB 窗口中输入：[V, D] = eig(B, 'nobalance')，可得到矩阵 B 的未进行平衡处理的特征值矩阵和模态矩阵：

```
V =
   -0.6153    0.4176    0.0000   -0.3540
    0.7881    0.3261         0   -0.3159
    0.0000    0.0000   -0.0000   -0.8715
   -0.0189   -0.8481   -1.0000   -0.1239
D =
    5.5616         0         0         0
         0    1.4384         0         0
         0         0    1.0000         0
         0         0         0   -1.0000
```

可以看出，两种方法求得的模态矩阵是不同的。为了验证，在 MATLAB 窗口中输入：B*V -V*D，可得：

```
ans =
  1.0e-015 *
    0.8882   -0.2220    0.1471   -0.2220
         0    0.0555   -0.3629    0.2776
   -0.0172   -0.0015    0.0066         0
    0.0278   -0.2220   -0.2220    0.1388
```

从上例可以看出，有时若不选用 nobalance 选项，计算结果将完全错误。这是由于调用 eig 函数时，首先要调用使各矩阵元素大致相当的平衡程序，该程序将把原矩阵中本可以忽

略的元素进行了放大。当 B 矩阵来自程序的中间计算结果时，就有可能发生本例的情况，但一般来说，nobalance 选项的作用是减少计算误差。

考虑下面两个方阵 A 和 B：

```
A =
     1     2     3
     4     5     6
     9     3     7
B =
    11    45    23
    14    23    15
    17    10    14
```

在 MATLAB 窗口中输入：d = eig(A,B)，可得一个包含广义特征值的向量 d：

```
d =
    0.5793
   -0.2966
   -0.1424
```

在 MATLAB 窗口中输入：[V, D] = eig(A, B)，可得一个广义特征值对角矩阵 D 和一个包含相应广义特征向量的模态矩阵 V：

```
V =
    0.3283    0.4716    0.4402
   -0.4333    0.3787    0.4469
    0.8393   -0.7964   -0.7788
D =
    0.5793         0         0
         0   -0.2966         0
         0         0   -0.1424
```

4. 广义奇异值分解

名称：gsvd

广义奇异值分解。

语法：该函数有如下几种表达形式：

- [U, V, X, C, S] = gsvd (A, B)
- [U, V, X, C, S] = gsvd (A, B, 0)
- sigma = gsvd (A, B)

描述：[U, V, X, C, S] = gsvd (A, B)返回单一化矩阵 U 和 V，一个方阵 X 和非负的对角矩阵 C 与 S，并满足：

A = U*C*X'

B = V*S*X'

C'*C + S'*S = I

矩阵 A 和 B 必须有相同的列数，但可以有不同的行数。如果 A 是 m×p 矩阵、B 是 n×p

矩阵，那么 U 将是 m×m 矩阵、V 将是 n×n 矩阵、X 将是 p×q 矩阵，其中 q = min(m+n,p)。

当 m≥p 或 n≥p 时，[U, V, X, C, S] = gsvd (A, B, 0)，将返回一个紧凑的分解形式。其中 U 阵和 V 阵最多有 p 列，C 阵和 S 阵最多有 p 行。并且广义奇异值为：diag(C)./diag(S)。

sigma = gsvd (A, B)返回广义奇异值 sqrt(diag(C'*C)./diag(S'*S))组成的向量。S 矩阵的非零元素常位于主对角线上。当 m≥p，则 C 矩阵的非零元素一般也位于主对角线上。但如果 m<p，则 C 矩阵的非零对角线为 diag(C,p-m)。

当 B 是一个方阵而且不奇异，则广义奇异值 gsvd(A,B)将等于普通的奇异值 svd(A/B)，但它们将按相反的顺序进行排列。

举例:

在下面的第一个例子中，A 和 B 矩阵的行数要大于(至少要等于)列数。其中：

A = reshape(1:15,5,3)，B = magic(3)，即：

```
A =
     1     6    11
     2     7    12
     3     8    13
     4     9    14
     5    10    15
B =
     8     1     6
     3     5     7
     4     9     2
```

在 MATLAB 窗口中输入：[U, V, X, C, S] = gsvd (A, B)，可得到 5×5 正交矩阵 U，3×3 正交矩阵 V，3×3 非奇异矩阵 X，和非负的对角矩阵 C 与 S：

```
U =
    0.5912   -0.6457   -0.4279    0.0998    0.2013
   -0.7797   -0.3296   -0.4375   -0.0009    0.3034
   -0.0084   -0.0135   -0.4470   -0.5781   -0.6825
   -0.0089    0.3026   -0.4566    0.7597   -0.3503
    0.2058    0.6187   -0.4661   -0.2805    0.5281
V =
   -0.7071    0.6946    0.1325
    0.0000    0.1874   -0.9823
    0.7071    0.6946    0.1325
X =
   -2.8284    9.3761   -6.9346
    5.6569    8.3071  -18.3301
   -2.8284    7.2381  -29.7256
C =
    0.0000         0         0
```

```
         0    0.3155         0
         0         0    0.9807
         0         0         0
         0         0         0
S =
    1.0000         0         0
         0    0.9489         0
         0         0    0.1957
```

由于 A 矩阵不是满秩的，C 矩阵的第一个对角元素为 0。

为了得到一个紧凑的分解形式，在 MATLAB 窗口中输入：[U,V,X,C,S] = gsvd(A,B,0)，可得：

```
U =
   -0.3736   -0.6457   -0.4279
   -0.0076   -0.3296   -0.4375
    0.8617   -0.0135   -0.4470
   -0.2063    0.3026   -0.4566
   -0.2743    0.6187   -0.4661
V =
   -0.7071    0.6946    0.1325
    0.0000    0.1874   -0.9823
    0.7071    0.6946    0.1325
X =
   -2.8284    9.3761   -6.9346
    5.6569    8.3071  -18.3301
   -2.8284    7.2381  -29.7256
C =
    0.0000         0         0
         0    0.3155         0
         0         0    0.9807
S =
    1.0000         0         0
         0    0.9489         0
         0         0    0.1957
```

为了求解广义奇异值组成的向量，在 MATLAB 窗口中输入：sigma = gsvd (A, B)，可得：

```
sigma =
    0.0000
    0.3325
    5.0123
```

从上可以看出，广义奇异值实际上矩阵 C 和矩阵 S 的相应对角元素的比值。可以计算

普通广义奇异值 svd(A/B)为：

```
    5.0123
    0.3325
    0.0000
```

在本例中，由于 B 是一个方阵而且不奇异，因此广义奇异值 gsvd(A,B)将等于普通的奇异值 svd(A/B)，但它们将按相反的顺序进行排列。

在第二个例子中，A 和 B 矩阵的列数要大于(至少要等于)行数。其中：

A = reshape(1:15,3,5)，B = magic(5)，即：

```
A =
     1     4     7    10    13
     2     5     8    11    14
     3     6     9    12    15
B =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

在 MATLAB 窗口中输入[U, V, X, C, S] = gsvd (A, B)，可得到 3×3 正交矩阵 U，5×5 正交矩阵 V，5×5 非奇异矩阵 X，和非负的对角矩阵 C 与 S：

```
U =
   -0.4082   -0.7178    0.5639
    0.8165   -0.0109    0.5772
   -0.4082    0.6961    0.5906
V =
    0.7776    0.0223    0.4250    0.4552    0.0843
    0.1416   -0.1382   -0.8120    0.5408   -0.0956
   -0.0712   -0.1385   -0.0866    0.0271    0.9836
   -0.4974   -0.5543    0.3790    0.5408   -0.0956
   -0.3505    0.8087    0.0945    0.4552    0.0843
X =
    7.3597         0   -6.9666   30.7287    4.6958
    6.6639    6.9182   11.8704   28.5003   11.6858
  -17.3706    6.9387    3.1805   26.2720   18.6758
   -4.3686  -14.5494   -1.5507   24.0436   25.6657
    7.7156    0.6925   -6.5336   21.8152   32.6557
C =
         0         0    0.0000         0         0
         0         0         0    0.0439         0
```

```
        0         0         0         0    0.7432
S =
   1.0000         0         0         0         0
        0    1.0000         0         0         0
        0         0    1.0000         0         0
        0         0         0    0.9990         0
        0         0         0         0    0.6690
```

广义奇异值 sigma = gsvd(A,B)含有 3 个 0：

```
sigma =
        0
        0
   0.0000
   0.0439
   1.1109
```

若将矩阵 A 和矩阵 B 的位置互换后再求广义奇异值，将产生 3 个无穷大值，如下所示：

```
gsvd(B,A)
ans =
   0.9001
  22.7610
      Inf
      Inf
      Inf
```

5. 矩阵的 Hessenvberg 形式

名称：hess

求解矩阵的 Hessenvberg 形式。

语法：该函数有如下两种表达形式：

- [P, H] = hess(A)
- H = hess(A)

描述：H = hess(A)返回矩阵 A 的 Hessenvberg 形式。

[P, H] = hess(A)返回一个 Hessenvberg 矩阵 H 和一个单位正交矩阵 P，并满足：

A = P*H*P'和 P'*P = eye(size(A))。

举例：

在下面的例子中，A 是一个 3×3 特征值测试矩阵：

```
A =
  -149   -50  -154
   537   180   546
   -27    -9   -25
```

在 MATLAB 窗口中输入：H = hess(A)，可得：

```
H =
```

```
 -149.0000    42.2037 -156.3165
 -537.6783   152.5511 -554.9272
         0     0.0728    2.4489
```

由上可见，矩阵 A 的 Hessenvberg 形式只在位置(3, 1)处为 0 。

在 MATLAB 窗口中输入：[P, H] = hess(A)，可得 Hessenvberg 矩阵 H 和单位正交矩阵 P：

```
P =
    1.0000         0         0
         0   -0.9987    0.0502
         0    0.0502    0.9987
H =
 -149.0000    42.2037 -156.3165
 -537.6783   152.5511 -554.9272
         0     0.0728    2.4489
```

6. 已知根的多项式的表达式

名称：poly

求已知根的多项式的表达式。

语法：该函数有如下两种表达形式：

- p = poly(A)
- p = poly(r)

描述：MATLAB 中多项式用行向量表示。例如多项式 $P(x)=a_0x^n+a_1x^{n-1}+...+a_{n-1}x+a_n$ 可以用系数行向量 $P=[a_1\quad a_2\quad ...\quad a_{n-1}\quad a_n]$ 表示。除了直接输入法，多项式行向量还可以由命令 poly 创建。

如果 A 是一个 n×n 矩阵，则 p = poly(A)将返回一个含有 n+1 个元素的行向量，这 n+1 个元素是特征多项式 det(sI - A)的系数，并且按降幂排列。

当 r 是一个向量时，p = poly(r)将返回一个行向量，该行向量的元素是以 r 中的元素为根的多项式的系数。

举例：

对于矩阵

```
A =
     1     2     3
     4     5     6
     7     8     0
```

在 MATLAB 窗口中输入 p = poly(A)，将得到一个含有 4 个元素(按降幂排列)的行向量：p = 1.0000 -6.0000 -72.0000 -27.0000。

即该多项式为：$x^4+x^3-6x^2-72x-27$。

对于向量 r = 12.1229 -5.7345 -0.3884，在 MATLAB 窗口中输入：p = poly(r)，将同样得到一个含有 4 个元素(按降幂排列)的行向量：

p = 1.0000 -6.0000 -72.0000 -27.0000。

7. QZ 分解

名称：qz

广义特征值的 QZ 分解。

语法：[AA, BB, Q, Z, V] = qz (A, B)

描述：[AA, BB, Q, Z, V] = qz (A, B)将返回上三角矩阵 AA 和 BB，左转换矩阵 Q 和右转换矩阵 Z，并满足：Q*A*Z = AA 和 Q*B*Z = BB，同时还要返回一个广义特征向量矩阵 V。其中 A 阵和 B 阵为方阵。

广义特征值是矩阵 AA 和矩阵 BB 的对角元素，并满足：A*V*diag(BB) = B*V*diag(AA)。

举例：

对于两个方阵 A 和 B：

```
A =
     1     2     3
     4     5     6
     9     3     7
B =
    12    20    15
    13     9    10
    23    19     7
```

为了进行广义特征值的 QZ 分解，在 MATLAB 窗口中输入[AA, BB, Q, Z, V] = qz (A, B)，可得：

```
AA =
  -1.2219 - 4.9838i  -5.7963 + 4.2495i  -5.9096 - 8.2581i
  -0.0000 - 0.0000i   1.1810 - 3.5533i   4.6433 - 3.2978i
  -0.0000 - 0.0000i  -0.0000 + 0.0000i   1.5563 - 0.1255i
BB =
   8.0608 -11.0544i  -6.6802 + 9.2272i -16.9214 -26.6389i
        0            -5.2147 - 8.5132i  15.5185 -14.8307i
        0                 0            -13.5447 + 1.0921i
Q =
  -0.1311 + 0.2185i  -0.0046 + 0.4053i   0.5731 + 0.6651i
  -0.3150 + 0.5471i  -0.3839 + 0.5012i  -0.1422 - 0.4274i
   0.7272 + 0.0876i  -0.6564 - 0.0791i   0.1611 + 0.0194i
Z =
  -0.2110 - 0.6396i   0.3603 + 0.5771i  -0.2833 + 0.0575i
   0.2648 - 0.0436i   0.0493 - 0.4167i  -0.8498 + 0.1726i
  -0.4157 + 0.5491i  -0.3622 + 0.4795i  -0.3975 + 0.0807i
V =
  -0.2110 - 0.6396i   0.5848 + 0.3341i  -0.1669 + 0.0339i
   0.2648 - 0.0436i   0.0955 - 0.2508i  -0.6709 + 0.1362i
```

-0.4157 + 0.5491i　-0.6208 + 0.2981i　0.6945 - 0.1410I

8. 实的 Schur 形式转换为复的 Schur 形式

名称：rsf2csf

把实的 Schur 形式转换为复的 Schur 形式。

语法：[U, T] = rsf2csf (U, T)

描述：矩阵的复的 Schur 形式是一个对角元素为该矩阵特征值的上三角矩阵。实的 Schur 形式在对角线上含有实的特征值和 2×2 复特征值子矩阵。

[U, T] = rsf2csf (U, T)把实的 Schur 形式转换为复的 Schur 形式。其中 U 是一个单位正交矩阵，T 是矩阵 A 的 Schur 形式，并满足如下条件：

A = U*T*U'

U'*U = eye(size(A))

详细内容可参见函数 schur 的说明。

举例：

对于给定的矩阵 A

```
A =
     1     1     1     3
     1     2     1     1
     1     1     3     1
    -2     1     1     4
```

利用函数 eig 可以求得矩阵 A 的特征值为：

1.9202-1.4742i　1.9202+1.4742i　4.8121　1.3474

首先利用函数[U, T]=schur(A)生成矩阵 A 的 Schur 形式：

```
U =
    0.4931    0.7993   -0.0197   -0.3428
    0.5762   -0.0082    0.1666    0.8001
    0.3780   -0.3981    0.7191   -0.4260
   -0.5310    0.4500    0.6743    0.2466
T =
    2.5650    1.4670    1.6104   -0.3020
   -1.7649    1.2754    2.7393    1.7569
         0         0    4.8121    1.1314
         0         0   -0.0000    1.3474
```

再利用函数[U, T] = rsf2csf (U, T)将其转换为复的 Schur 形式：

```
U =
  -0.4576 + 0.3044i    0.5802 - 0.4934i   -0.0197   -0.3428
   0.1616 + 0.3556i    0.4235 + 0.0051i    0.1666    0.8001
   0.3963 + 0.2333i    0.1718 + 0.2458i    0.7191   -0.4260
  -0.4759 - 0.3278i   -0.2709 - 0.2778i    0.6743    0.2466
T =
   1.9202 + 1.4742i    0.7691 - 1.0772i   -1.5895 - 0.9940i   -1.3798 + 0.1864i
```

0	1.9202 - 1.4742i	1.9296 + 1.6909i	0.2511 + 1.0844i
0	0	4.8121	1.1314
0	0	0	1.3474

9. Schur 分解

名称：schur

Schur 分解。

语法：该函数有如下两种表达形式：

- [U, T] = schur (A)
- T = schur (A)

描述：矩阵的复的 Schur 形式是一个对角元素为该矩阵特征值的上三角矩阵。实的 Schur 形式在对角线上含有实的特征值和 2×2 复特征值子矩阵。

schur 函数用来计算矩阵的 Schur 形式。如果矩阵是实的，schur 函数将返回实的 Schur 形式；如果矩阵是复的，则返回一个复的 Schur 形式。函数 rsf2csf 可以将实的 Schur 形式转换为复的 Schur 形式。

[U, T] = schur (A)返回一个 Schur 矩阵 T 和一个单位正交矩阵 U，并满足：

A = U*T*U'

U'*U = eye (size (A))

其中 A 必须为方阵。

T = schur (A)只返回一个 Schur 矩阵 T。

举例：

A 是一个 3×3 特征值测试矩阵：

```
A =
  -149   -50  -154
   537   180   546
   -27    -9   -25
```

在 MATLAB 窗口中输入[U, T] = schur (A)，可得到 Schur 矩阵 T 和单位正交矩阵 U：

```
U =
    0.3162    0.6529   -0.6882
   -0.9487    0.2176   -0.2294
    0.0000   -0.7255   -0.6882
T =
    1.0000    7.1119  815.8706
         0    2.0000  -55.0236
         0         0    3.0000
```

若在 MATLAB 窗口中输入 T = schur (A)，则只得到 Schur 矩阵 T。

10. 奇异值分解

名称：svd

奇异值分解。

语法：该函数有如下几种表达形式：

- s = svd (X)
- [U, S, V] = svd (X)
- [U, S, V] = svd (X, 0)

描述：奇异值分解在矩阵分析具有极为重要的作用。奇异值分解的定义为：对于 m×n(不失一般性，设 m>n)阶矩阵 X，若存在 m×m 阶矩阵 U 和 n×n 阶矩阵 V，使得 $X=U*W*V^T$，则称 U、W 和 V 为矩阵 X 的奇异值分解三对组。

MATLAB 中，奇异值分解是通过 svd 函数来实现的。

s = svd (X)返回一个奇异值向量。

[U, S, V] = svd (X)返回一个和矩阵 X 同维数的对角矩阵 S，并且 S 阵中的非负对角元素按降序排列，还返回单位正交矩阵，并满足：X = U*S*V'。

[U, S, V] = svd (X, 0)进行紧凑型分解。如果 X 是 m×n (并且 m>n)阶矩阵，则函数 svd 只计算 U 矩阵的前 n 列，而且 S 是一个 n×n 阶矩阵。

举例：

对如下矩阵 X 进行普通奇异值分解和紧凑形式的奇异值分解：

```
X =
     9     4
     6     8
     2     7
```

在 MATLAB 窗口中输入[U, S, V] = svd (X)，可得到矩阵 A 的普通奇异值分解：

```
U =
    0.6105   -0.7174    0.3355
    0.6646    0.2336   -0.7098
    0.4308    0.6563    0.6194
S =
   14.9359         0
         0    5.1883
         0         0
V =
    0.6925   -0.7214
    0.7214    0.6925
```

在 MATLAB 窗口中输入[U, S, V] = svd (X, 0)，可得到矩阵 A 的紧凑形式的奇异值分解：

```
U =
    0.6105   -0.7174
    0.6646    0.2336
    0.4308    0.6563
S =
   14.9359         0
         0    5.1883
V =
```

```
0.6925   -0.7214
0.7214    0.6925
```

6.4 矩阵函数

1. 矩阵指数

名称：expm

矩阵指数。

语法：Y = expm (X)

描述：MATLAB 计算矩阵指数有三种方法：

1，第一种方法是采用 MATLAB 中的内置函数 expm (该方法的说明存储在 M 文件 expm2.m 中)。

2，第二种方法是采用 Taylor 系列近似法(该方法的说明存储在 M 文件 expm2.m 中)。该方法的缺点是计算速度比较慢，而且精度较差。

3，第三种方法首先将矩阵对角化，然后对每一个独立的特征值调用 exp 函数，最后再将其变换为一般形式的矩阵。

Y = expm (X)返回矩阵 X 的指数矩阵。如果矩阵 X 含有非正的特征值，则会得到复数结果。

函数 exp (X)可以用来求解矩阵 X 中每个元素的指数形式。

举例：

例如对如下所示一个 3×3 阶的矩阵：

```
X =
     1     1     0
     0     0     2
     0     0    -1
```

在 MATLAB 窗口中输入 Y = expm (X)，可得到矩阵 X 的指数矩阵：

```
Y =
    2.7183    1.7183    1.0862
         0    1.0000    1.2642
         0         0    0.3679
```

若输入 Y = exp (X)，可得到矩阵 X 中每个元素的指数形式：

```
Y =
    2.7183    2.7183    1.0000
    1.0000    1.0000    7.3891
    1.0000    1.0000    0.3679
```

从上述的结果中可以看出，两个矩阵中的对角线元素是一致的。这个结论对任何一个三角形矩阵都是正确的。但非对角元素是不同的。

2. 一般矩阵函数

名称：funm

计算一般矩阵函数。

语法：该函数有如下两种表达形式：

- Y = funm (X, 'function')
- [Y, esterr] = funm (X, 'function')

描述：Y = funm (X, 'function')采用 Parlett 法计算矩阵函数。其中 X 必须为方阵。

命令 funm (X, 'sqrt')和命令 funm (X, 'log')与命令 sqrtm(X)和 logm(X)是等效的。命令 funm(X,'exp')和命令 expm(X)能够计算得到相同的函数，但是采用不同的方法。一般情况下，推荐使用 expm(X)。

[Y, esterr] = funm (X, 'function')不会输出任何信息，但会返回一个对计算结果的相对误差的一个粗略估计。如果 X 是一个对称矩阵或 Hermitian 矩阵，则其 Schur 形式为对角矩阵，此时采用 funm 就可以得到一个精确的结果。

举例：

在 MATLAB 窗口中输入 S = funm(X,'sin')，可得到矩阵函数：

```
S =
    0.8415    0.8415         0
         0         0    1.6829
         0         0   -0.8415
```

在 MATLAB 窗口中输入 C = funm(X,'cos')，可得到矩阵函数：

```
C =
    0.5403   -0.4597   -0.9194
         0    1.0000    0.9194
         0         0    0.5403
```

上面的结果与下面操作的结果相同：

在 MATLAB 窗口中输入 E = expm(i*X)，可得到：

```
E =
   0.5403 + 0.8415i   -0.4597 + 0.8415i   -0.9194 - 0.0000i
        0                  1.0000                 0.9194 + 1.6829i
        0                       0                 0.5403 - 0.8415I
```

然后在 MATLAB 窗口中输入 C = real(E)和 S = imag(E)，可分别得到：

```
C =
    0.5403   -0.4597   -0.9194
         0    1.0000    0.9194
         0         0    0.5403
S =
    0.8415    0.8415   -0.0000
         0         0    1.6829
         0         0   -0.8415
```

可以看出两种方法得到的结果完全相同。

在 MATLAB 窗口中输入[Y, esterr] = funm (X, 'sin')和[Z, esterr] = funm (X, 'cos')，可分别得到：

```
Y =
    0.8415    0.8415         0
         0         0    1.6829
         0         0   -0.8415
esterr = 4.4409e-016
Z =
    0.5403   -0.4597   -0.9194
         0    1.0000    0.9194
         0         0    0.5403
esterr = 4.4409e-016
```

3. 矩阵对数

名称：logm

矩阵对数。

语法：该函数有如下两种表达形式：

- Y = logm (X)
- [Y, esterr] = logm (X)

描述：Y = logm (X)返回矩阵对数，即 expm(X)的逆函数。如果 X 有负的特征值，则会得到复数结果。若 expm(Y)的计算结果不与 X 接近，则会输出一个警告信息。

[Y, esterr] = logm (X)不会输出任何警告信息，但会返回一个对 norm(expm(Y)-X)/norm(X)的相对残差的估计。

举例：

考虑如下所示的一个 3x3 矩阵：

```
A =
     1     1     0
     0     0     2
     0     0    -1
```

首先在 MATLAB 窗口中输入 X = expm(A)，可得到矩阵指数：

```
X =
    2.7183    1.7183    1.0862
         0    1.0000    1.2642
         0         0    0.3679
```

然后在输入 Y = logm (X)，得到原始矩阵：

```
Y =
    1.0000    1.0000    0.0000
         0         0    2.0000
         0         0   -1.0000
```

可以看出矩阵 A 与矩阵 Y 完全相同。

在 MATLAB 窗口中输入[Y, esterr] = logm (X)，可得：

```
Y =
    1.0000    1.0000    0.0000
         0         0    2.0000
         0         0   -1.0000
esterr = 8.2562e-016
```

4. 矩阵平方根

名称：sqrtm

矩阵平方根。

语法：该函数有如下两种表达形式：

- Y = sqrtm (X)
- [Y, esterr] = sqrtm (X)

描述：Y = sqrtm (X)返回矩阵 X 的矩阵平方根。如果 X 有负的特征值，则会得到复数结果。若 Y*Y 的计算结果不与 X 接近，则会输出一个警告信息。

[Y, esterr] = sqrtm (X)不会输出任何警告信息，但会返回一个对 norm(Y*Y-X)/norm(X)的相对残差的估计。

举例：

考虑如下所示的对称、正定矩阵：

```
X =
     5    -4     1     0     0
    -4     6    -4     1     0
     1    -4     6    -4     1
     0     1    -4     6    -4
     0     0     1    -4     5
```

在 MATLAB 窗口中输入 Y = sqrtm (X)，可得到 X 的矩阵平方根：

```
Y =
    2.0000   -1.0000    0.0000    0.0000    0.0000
   -1.0000    2.0000   -1.0000         0   -0.0000
    0.0000   -1.0000    2.0000   -1.0000   -0.0000
    0.0000         0   -1.0000    2.0000   -1.0000
    0.0000   -0.0000   -0.0000   -1.0000    2.0000
```

在 MATLAB 窗口中输入[Y, esterr] = sqrtm (X)，可得：

```
Y =
    2.0000   -1.0000    0.0000    0.0000    0.0000
   -1.0000    2.0000   -1.0000         0   -0.0000
    0.0000   -1.0000    2.0000   -1.0000   -0.0000
    0.0000         0   -1.0000    2.0000   -1.0000
    0.0000   -0.0000   -0.0000   -1.0000    2.0000
```

esterr = 2.2204e-016

6.5 低级函数

1. 从 QR 分解中删除列

名称：qrdelete

从 QR 分解中删除列。

语法：[Q, R] = qrdelete (Q, R, j)

描述：[Q, R] = qrdelete (Q, R, j) 返回矩阵 A 去掉第 j 列(即 A(:, j))后再进行 QR 分解得到的矩阵。

函数的输入参量 Q 和 R 是矩阵 A 经过 QR 分解(利用函数[Q,R] = qr(A))得到的原始矩阵。参数 j 指从矩阵 A 中去掉的列。

举例：

对于矩阵

```
A =
     1     2     3
     4     5     6
     7     8     9
    10    11    12
```

首先对其进行 QR 分解，即在 MATLAB 命令窗口中输入：[Q, R] = qr(A)，可得：

```
Q =
   -0.0776   -0.8331   -0.2636   -0.4801
   -0.3105   -0.4512    0.7093    0.4437
   -0.5433   -0.0694   -0.6278    0.5530
   -0.7762    0.3124    0.1821   -0.5166
R =
  -12.8841  -14.5916  -16.2992
         0   -1.0413   -2.0826
         0         0   -0.0000
         0         0         0
```

去掉矩阵 A 中的第一列，然后再进行 QR 分解。即在 MATLAB 命令窗口中输入：

[Q, R] = qrdelete (Q, R, 1)

会得到如下所示的结果：

```
Q =
    0.1367    0.8254   -0.2636   -0.4801
    0.3418    0.4280    0.7093    0.4437
    0.5469    0.0306   -0.6278    0.5530
    0.7519   -0.3669    0.1821   -0.5166
```

```
R =
   14.6287   16.4061
         0    0.9171
         0         0
         0         0
```

2. 在 QR 分解中加入列

名称：qrinsert

在 QR 分解中加入列。

语法：[Q, R] = qrinsert (Q, R, j, x)

描述：[Q, R] = qrinsert (Q, R, j, x)返回矩阵 A 在第 j 列(即 A(:, j))前插入一个外部向量 x，再进行 QR 分解得到的矩阵。如果矩阵 A 有 n 列，而 j=n+1，则 qrinsert 在最后一列后插入 x。

函数的输入参量 Q 和 R 是矩阵 A 经过 QR 分解(利用函数[Q,R] = qr(A))得到的原始矩阵。参数 j 指从矩阵 A 中去掉的列。

举例：

对于如下所示的矩阵 A 和向量 x：

```
A =
     1     2     3
     4     5     6
     7     8     9
    10    11    12
x =
     3
     6
     4
     5
```

首先对 A 进行 QR 分解，即在 MATLAB 命令窗口中输入：[Q, R] = qr(A)，可得：

```
Q =
   -0.0776   -0.8331   -0.2636   -0.4801
   -0.3105   -0.4512    0.7093    0.4437
   -0.5433   -0.0694   -0.6278    0.5530
   -0.7762    0.3124    0.1821   -0.5166
R =
  -12.8841  -14.5916  -16.2992
         0   -1.0413   -2.0826
         0         0   -0.0000
         0         0         0
```

然后在矩阵 A 的第 2 列前插入向量 x，再进行 QR 分解。即在 MATLAB 命令窗口中输入：[Q, R] = qrinsert (Q, R, 2, x)，此时会得到如下所示的结果：

```
Q =
    0.1367    0.4730   -0.8065   -0.3273
    0.3418    0.7991    0.4824    0.1091
    0.5469   -0.1529   -0.3070    0.7638
    0.7519   -0.3380    0.1506   -0.5455
R =
   14.6287    8.4081   16.4061
         0    3.9120    0.7812
         0         0   -0.4805
         0         0         0
```

第七章　数据分析和傅里叶变换

7.1　基本运算

1. 凸壳函数

名称：convhull

凸壳函数。

语法：该函数有如下两种表达形式：

- K = convhull (x, y)
- K = convhull (x, y, TRI)

描述：K = convhull (x, y)返回凸壳上点的向量 x 和 y 的索引值。

K = convhull (x, y, TRI)使用三角化分析(利用函数 delaunay)，而不是每一次都进行计算。

举例：

考虑向量 xx = -1:.05:1，即：

```
xx =
  Columns 1 through 7
   -1.0000    -0.9500    -0.9000    -0.8500    -0.8000    -0.7500    -0.7000
  Columns 8 through 14
   -0.6500    -0.6000    -0.5500    -0.5000    -0.4500    -0.4000    -0.3500
  Columns 15 through 21
   -0.3000    -0.2500    -0.2000    -0.1500    -0.1000    -0.0500          0
  Columns 22 through 28
    0.0500     0.1000     0.1500     0.2000     0.2500     0.3000     0.3500
  Columns 29 through 35
    0.4000     0.4500     0.5000     0.5500     0.6000     0.6500     0.7000
  Columns 36 through 41
    0.7500     0.8000     0.8500     0.9000     0.9500     1.0000
```

在 MATLAB 命令窗口中输入：yy = abs(sqrt(xx))，得到向量 yy：

```
yy =
  Columns 1 through 7
    1.0000     0.9747     0.9487     0.9220     0.8944     0.8660     0.8367
  Columns 8 through 14
    0.8062     0.7746     0.7416     0.7071     0.6708     0.6325     0.5916
  Columns 15 through 21
```

```
    0.5477    0.5000    0.4472    0.3873    0.3162    0.2236         0
  Columns 22 through 28
    0.2236    0.3162    0.3873    0.4472    0.5000    0.5477    0.5916
  Columns 29 through 35
    0.6325    0.6708    0.7071    0.7416    0.7746    0.8062    0.8367
  Columns 36 through 41
    0.8660    0.8944    0.9220    0.9487    0.9747    1.0000
```

再在 MATLAB 命令窗口中输入[x,y] = pol2cart(xx,yy)，把极坐标转换为直角坐标：

```
x =
  Columns 1 through 7
    0.5403    0.5670    0.5897    0.6085    0.6232    0.6337    0.6399
  Columns 8 through 14
    0.6418    0.6393    0.6322    0.6205    0.6040    0.5825    0.5557
  Columns 15 through 21
    0.5233    0.4845    0.4383    0.3829    0.3146    0.2233         0
  Columns 22 through 28
    0.2233    0.3146    0.3829    0.4383    0.4845    0.5233    0.5557
  Columns 29 through 35
    0.5825    0.6040    0.6205    0.6322    0.6393    0.6418    0.6399
  Columns 36 through 41
    0.6337    0.6232    0.6085    0.5897    0.5670    0.5403
y =
  Columns 1 through 7
   -0.8415   -0.7928   -0.7431   -0.6926   -0.6416   -0.5903   -0.5390
  Columns 8 through 14
   -0.4879   -0.4374   -0.3876   -0.3390   -0.2918   -0.2463   -0.2029
  Columns 15 through 21
   -0.1619   -0.1237   -0.0888   -0.0579   -0.0316   -0.0112         0
  Columns 22 through 28
    0.0112    0.0316    0.0579    0.0888    0.1237    0.1619    0.2029
  Columns 29 through 35
    0.2463    0.2918    0.3390    0.3876    0.4374    0.4879    0.5390
  Columns 36 through 41
    0.5903    0.6416    0.6926    0.7431    0.7928    0.8415
```

然后在 MATLAB 命令窗口中输入 k = convhull(x,y)，得到凸壳上点的向量 x 和 y 的索引值：k =

```
  Columns 1 through 12
     1     2     3     4     5     6     7     8    34    35    36    37
  Columns 13 through 18
```

38　　39　　40　　41　　21　　1

为了更直观地了解计算结果，可以利用 MATLAB 提供的绘图函数将这些点绘制出来。在在 MATLAB 命令窗口中输入 plot(x(k),y(k),'r-',x,y,'b+')，可以得到如图 7-1 所示的曲线。

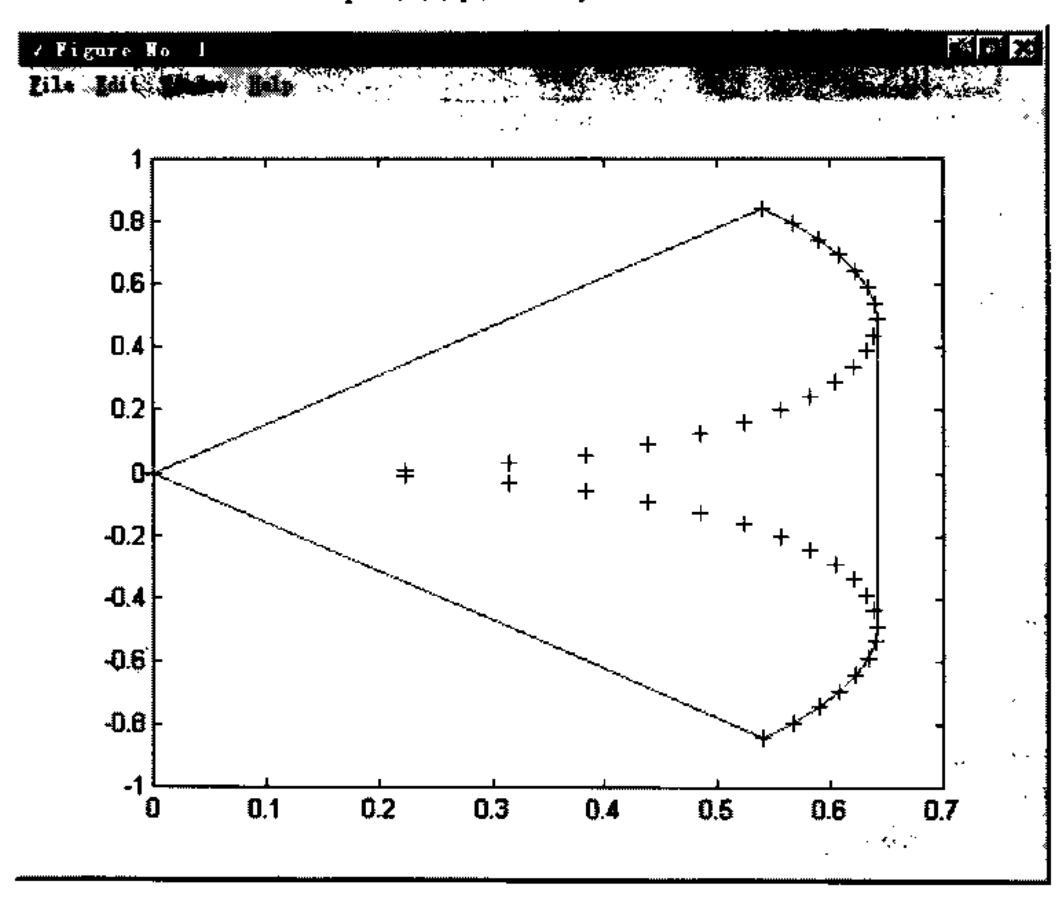


图 7-1　凸壳上的点

2. 累计积

名称：cumprod

累计积。

语法：该函数有如下两种表达形式：

- B = cumprod (A)
- B = cumprod (A, dim)

描述：B = cumprod (A)返回累计积。

如果 A 是一个向量，cumprod (A)将返回一个包含 A 中元素累计积的向量。

如果 A 是一个矩阵，cumprod (A)将返回一个和 A 同维的矩阵，该矩阵包含 A 矩阵的每一列的累计积。

B = cumprod (A, dim)返回 A 中沿着标量 dim 指定的维数的元素累计积。

举例：

例如对于如下所示的矩阵：

A =

　　1　　2　　3

```
    4    5    6
```

在 MATLAB 命令窗口中输入 B = cumprod (A)，得到矩阵 A 的累计积向量：

```
B =
    1    2    3
    4   10   18
```

在 MATLAB 命令窗口中输入 C = cumprod (A, 2)，

```
C =
    1    2    6
    4   20  120
```

3. 累计和

名称：cumsum

累计和。

语法：该函数有如下两种表达式：

- B = cumsum (A)
- B = cumsum (A, dim)

描述：B = cumsum (A)返回累计和。

如果 A 是一个向量，cumsum (A)将返回一个包含 A 中元素累计和的向量。

如果 A 是一个矩阵，cumsum (A)将返回一个和 A 同维的矩阵，该矩阵包含 A 矩阵的每一列的累计和。

B = cumsum (A, dim)返回 A 中沿着标量 dim 指定的维数的元素累计和。

举例：

例如对于如下所示的矩阵：

```
A =
    1    2    3
    4    5    6
```

在 MATLAB 命令窗口中输入 B = cumsum (A)，得到矩阵 A 的累计和向量：

```
B =
    1    2    3
    5    7    9
```

在 MATLAB 命令窗口中输入 C = cumsum (A, 2)，

```
C =
    1    3    6
    4    9   15
```

4. 累计梯形数值积分

名称：cumtrapz

计算累计梯形数值积分。

语法：该函数有如下几种表达形式：

- Z = cumtrapz (Y)
- Z = cumtrapz (X, Y)

- Z = cumtrapz (... dim)

描述：Z = cumtrapz (Y)返回采用梯形积分法计算得到的 Y 的累计积分的近似值。

如果 Y 是一个向量，cumtrapz (Y)将返回一个包含 Y 中元素累计积分的向量。

如果 Y 是一个矩阵，cumtrapz(Y)将返回一个包含各列累计积分的行向量。

Z = cumtrapz (X, Y)返回使用梯形积分法计算得到的与 X 相关的 Y 的累计积分。其中 X 和 Y 必须是同样长度的向量，或者 X 必须是一个列向量而 Y 是一个数组。

Z = cumtrapz (... dim)返回沿着 Y 的、由标量 dim 指定的维数的积分。X 的长度必须与 size(Y,dim)相同

举例：

例如对于如下所示的矩阵：

```
Y =
      0      1      2
      3      4      5
```

在 MATLAB 命令窗口中输入 Z = cumtrapz (Y)，得到矩阵 Y 的累计数值积分：

```
Z =
          0          0          0
     1.5000     2.5000     3.5000
```

在 MATLAB 命令窗口中输入 Z1 = cumtrapz (Y, 2)，可以得到：

```
Z1 =
          0     0.5000     2.0000
          0     3.5000     8.0000
```

5. Delaunay 三角化

名称：delaunay

Delaunay 三角化。

语法：该函数有如下两种表达形式：

- TRI = delaunay (x, y)
- TRI = delaunay (x, y, 'sorted')

描述：对于给定的一组数据点集，所谓 Delaunay 三角化，就是连接每一点及其相邻自然点的一组线段集。

TRI = delaunay (x, y)返回一组三角形集。而且所有这些三角形的外接圆都不包含数据点。m×3 矩阵 TRI 的每一行都包含一个这样的三角形，并且包含向量 x 和 y 的索引。为了避免共线数据的蜕化，delaunay 函数在数据中增加了随机模糊数。

TRI = delaunay (x, y, 'sorted')假设点 x 和 y 按照先 y 后 x 的顺序进行排列，并且假设重复点已经删除。

举例：

下面的例子将对随机产生的 10 个点进行 Delaunay 三角化。

首先利用函数 rand('state',0)设置随机数的产生方式。然后在 MATLAB 命令窗口中输入 x = rand(1,10)，可以得到 10 个随机数：

```
x =
```

```
  Columns 1 through 7
    0.9501    0.2311    0.6068    0.4860    0.8913    0.7621    0.4565
  Columns 8 through 10
    0.0185    0.8214    0.4447
```

再输入 y = rand(1,10)，得到另外 10 个随机数：

```
y =
  Columns 1 through 7
    0.6154    0.7919    0.9218    0.7382    0.1763    0.4057    0.9355
  Columns 8 through 10
    0.9169    0.4103    0.8936
```

利用函数 TRI = delaunay(x,y)进行 Delaunay 三角化，得到：

```
TRI =
     6     9     5
     1     5     9
     6     1     9
     2     4     6
     5     2     6
     4     1     6
     2    10     4
    10     3     4
     3     1     4
     7     3    10
     2     7    10
     5     8     2
     8     7     2
```

接着调用 MATLAB 提供的绘图函数 subplot(1,2,1)和三角形网格自动划分函数：trimesh(TRI,x,y,zeros(size(x)))将各个点连接成多个三角形，并利用函数 view(2)、axis([0 1 0 1])、hold on、plot(x,y,'o')和 set(gca,'box','on')设置图形显示的角度和方式及标签。最后的显示结果如图 7-2 中左面的图所示。

为了与由这些相同的点得到的 Voronoi 图进行比较，在 MATLAB 中输入[vx, vy] = voronoi(x,y,TRI)，可得：

```
vx =
  Columns 1 through 7
    0.7090    0.3636    0.7271   -0.1117    0.7785    0.2529    0.7090
    0.7271    0.2529    0.5384   -1.2073    0.7090   -0.1117    0.2529
  Columns 8 through 14
   -0.1117    0.2327   -1.2073    1.1134    0.8018    0.7785    0.2810
    0.8018    0.2810    0.2327    0.7785    1.1134    0.8018    0.3636
  Columns 15 through 17
```

```
    0.5384    0.3636    0.5284
    0.5284    0.5384    0.2810
vy =
  Columns 1 through 7
    0.6425    0.7889    0.7111   -0.2374    0.5802    0.2637    0.6425
    0.7111    0.2637    0.8353   -1.4122    0.6425   -0.2374    0.2637
  Columns 8 through 14
   -0.2374    1.0381   -1.4122    0.3700    0.2770    0.5802    0.9623
    0.2770    0.9623    1.0381    0.5802    0.3700    0.2770    0.7889
  Columns 15 through 17
    0.8353    0.7889    0.8927
    0.8927    0.8353    0.9623
```

然后调用 MATLAB 提供的绘图函数 subplot(1,2,2)和 plot(x,y,'r+',vx,vy,'b-')，以及显示角度的设置函数 axis([0 1 0 1])。最后的显示结果如图 7-2 中右面的图所示。

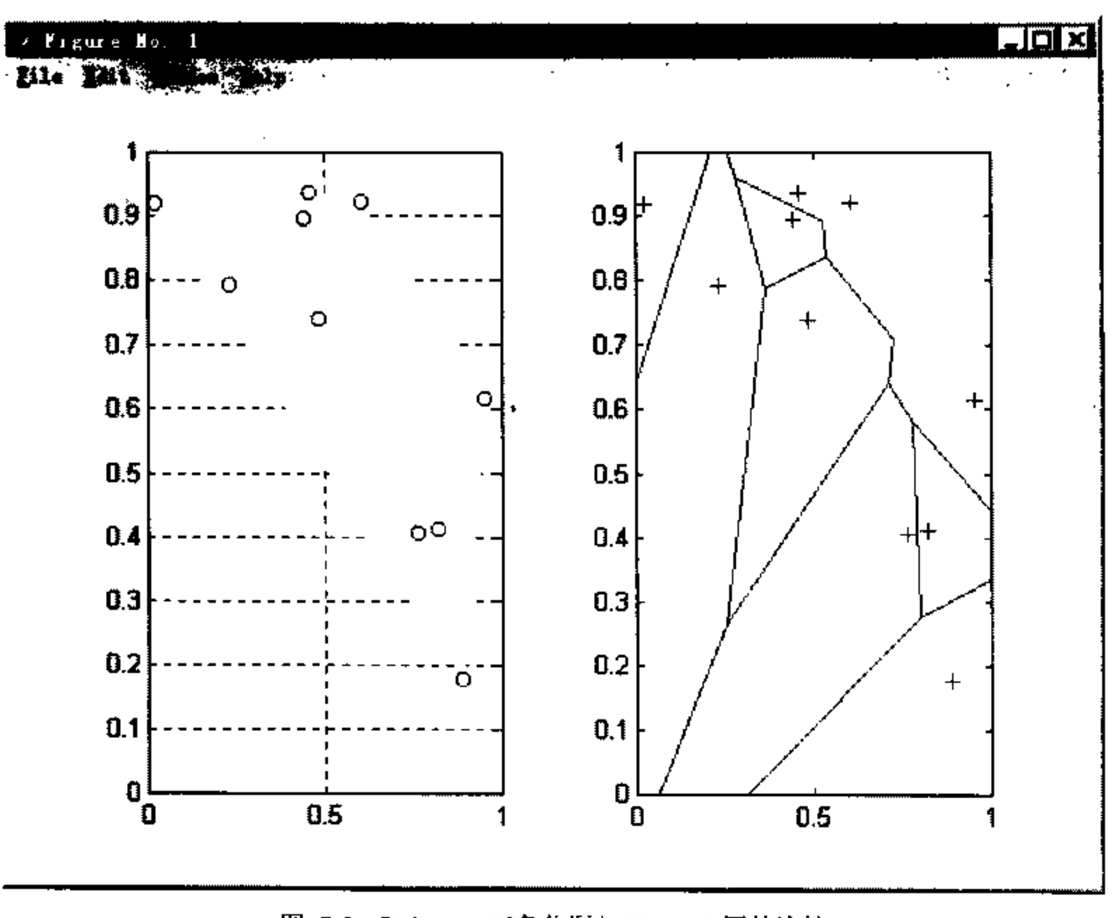


图 7-2 Delaunay 三角化图与 Voronoi 图的比较

6. 求最近点

名称：dsearch

求最近点。

语法：该函数有如下两种表达形式：

- K = dsearch (x, y, TRI, xi, yi)
- K = dsearch (x, y, TRI, xi, yi, S)

描述：K = dsearch (x, y, TRI, xi, yi)返回距离点(xi, yi)最近的点的索引。函数 dsearch 需要一个由函数 delaunay 生成的点(x, y)的 Delaunay 三角化矩阵 TRI。

K = dsearch (x, y, TRI, xi, yi, S)使用稀疏矩阵 S，而不需要每次进行计算。其中，S 为：

S = sparse(TRI(:,[1 1 2 2 3 3]),TRI(:,[2 3 1 3 1 2]),1,nxy,nxy)，nxy = prod(size(x))。

举例：

下面的例子将对随机产生的 10 个点进行 dsearch 运算。

首先利用函数 rand('state',1)设置随机数的产生方式。然后在 MATLAB 命令窗口中输入 x = rand(1,10)，可以得到 10 个随机数：

```
x =
  Columns 1 through 7
    0.9528    0.7041    0.9539    0.5982    0.8407    0.4428    0.8368
  Columns 8 through 10
    0.5187    0.0222    0.3759
```

再输入 y = rand(1,10)，得到另外 10 个随机数：

```
y =
  Columns 1 through 7
    0.8986    0.4290    0.1996    0.3031    0.5383    0.9102    0.5253
  Columns 8 through 10
    0.3068    0.0345    0.7153
```

利用函数 TRI = delaunay(x,y)进行 Delaunay 三角化，得到：

```
TRI =
     9     4     3
     8     4     9
     2     3     4
     7     3     2
     8     2     4
     2     5     7
     5     3     7
    10     2     8
     9    10     8
    10     5     2
     6     5    10
     6     1     5
     1     3     5
     9     6    10
```

接着在 MATLAB 命令窗口中输入 K = dsearch (x, y, TRI, 0, 0)，求解距离原点最近的点。结果为 k = 9，即点(0.0222, 0.0345)距离原点最近。

为了使用稀疏矩阵 S，在 MATLAB 命令窗口中首先输入 nxy = prod(size(x))，得到：nxy=10。然后输入 S = sparse(TRI(:,[1 1 2 2 3 3]),TRI(:,[2 3 1 3 1 2]),1,nxy,nxy)，得到一个稀疏矩阵：

```
S =
    (3,1)        1
    (5,1)        2
    (6,1)        1
    (3,2)        2
    (4,2)        2
    (5,2)        2
    (7,2)        2
    (8,2)        2
   (10,2)        2
    (1,3)        1
    (2,3)        2
    (4,3)        2
    (5,3)        2
    (7,3)        2
    (9,3)        1
    (2,4)        2
    (3,4)        2
    (8,4)        2
    (9,4)        2
    (1,5)        2
    (2,5)        2
    (3,5)        2
    (6,5)        2
    (7,5)        2
   (10,5)        2
    (1,6)        1
    (5,6)        2
    (9,6)        1
   (10,6)        2
    (2,7)        2
    (3,7)        2
    (5,7)        2
    (2,8)        2
    (4,8)        2
    (9,8)        2
```

(10,8)	2
(3,9)	1
(4,9)	2
(6,9)	1
(8,9)	2
(10,9)	2
(2,10)	2
(5,10)	2
(6,10)	2
(8,10)	2
(9,10)	2

最后输入K = dsearch (x, y, TRI, 1, 1, S)，可得k = 1。这表明距离点(1, 1)最近的点为(0.9528, 0.8986)。

7. 质数分解

名称：factor

质数分解。

语法：该函数有如下两种表达形式：

- f = factor(n)
- f = factor(symb)

描述：f = factor(n)返回一个包含 n 的质数分解因子的行向量。

举例：

取 n=123，在 MATLAB 命令窗口中输入 f = factor(n)，可以得到 n 的质数分解为：

f = 3　　41。

8. 搜索多边形内的点

名称：inpolygon

搜索多边形内的点。

语法：IN = inpolygon(X,Y,xv,yv)

描述：IN = inpolygon(X,Y,xv,yv)返回一个和 X、Y 同样大小的矩阵 IN。IN 中的每一个元素均取 1、0.5 或 0 中的一个值(根据点(X(p,q),Y(p,q))是否位于多边形内确定取值)。该多边形的顶点用向量 xv 和 yv 描述。具体来讲：

如果(X(p,q)，Y(p,q))位于多边形内，则 IN(p,q) = 1。

如果(X(p,q)，Y(p,q))位于多边形的边界上，则 IN(p,q) = 0.5。

如果(X(p,q)，Y(p,q))位于多边形外，则 IN(p,q) = 0。

举例：

下面的例子首先确定多边形的形状和边界，然后再随机地选取250个点，并利用inpolygon函数判断这些是否位于该多边形之内。具体操作为：

在 MATLAB 命令窗口中输入 L = linspace(0,2.*pi,6)，得到：

L =　0　　1.2566　　2.5133　　3.7699　　5.0265　　6.2832

输入 xv = cos(L)'和 yv = sin(L)确定多边形的顶点，得到顶点向量：

```
xv =                    yv =
    1.0000                   0
    0.3090                0.9511
   -0.8090                0.5878
   -0.8090               -0.5878
    0.3090               -0.9511
    1.0000               -0.0000
```

再输入 x = randn(250,1)和 y = randn(250,1)，产生 250 随机点。然后输入 in = inpolygon(x,y,xv,yv)搜索位于多边形内的点。并用函数 plot(xv,yv,x(in),y(in),'r+',x(~in),y(~in),'bo')将其图形化，如图 7-3 所示。

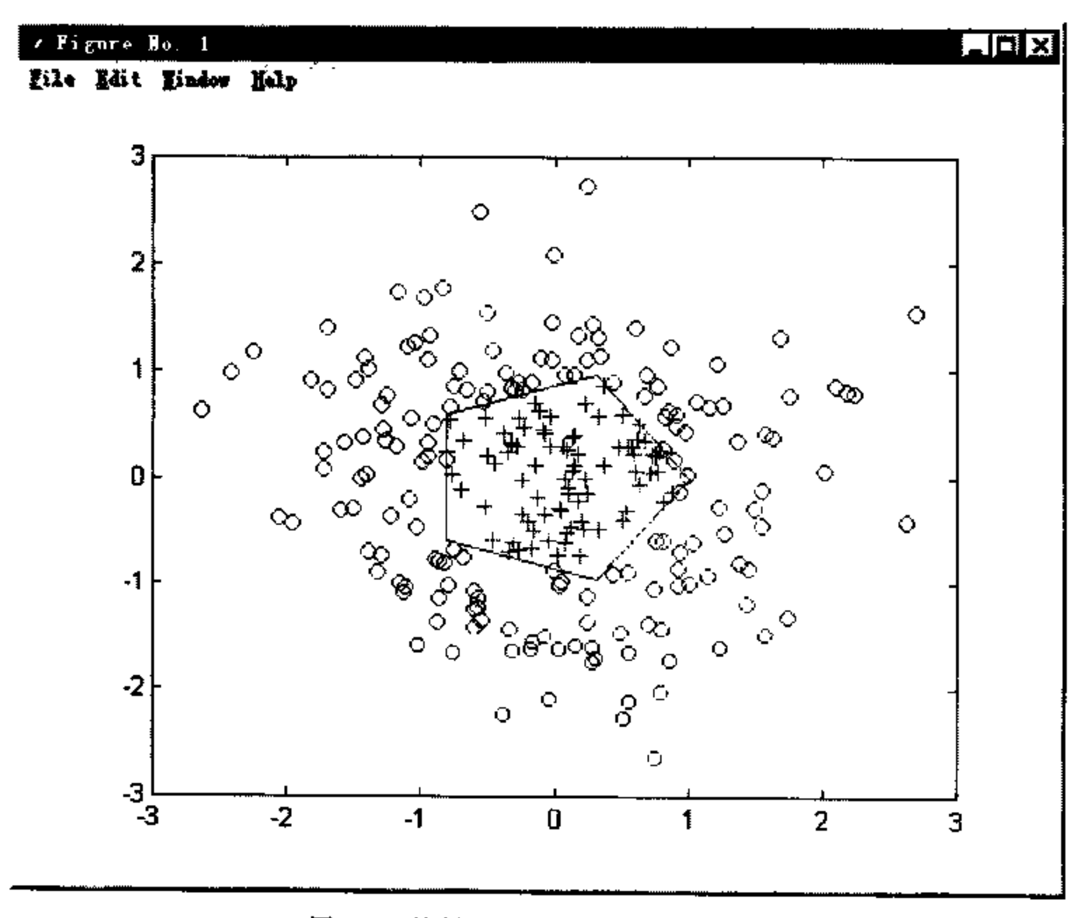


图 7-3　绘制 inpolygon 函数搜索的结果

9．求数组中的最大元素

名称：max

求数组中的最大元素。

语法：该函数有如下几种表达形式：

- C = max (A)
- C = max (A, B)
- C = max (A,[],dim)
- [C, I] = max(...)

描述：C = max (A)返回数组 A 中的最大元素。

如果 A 是一个向量，则 max (A)返回 A 中的最大元素。

如果 A 是一个矩阵，则 max (A)首先将 A 中的每一列作为一个向量，然后返回一个包含各个列向量中最大元素的行向量。

C = max (A, B)返回一个包含 A、B 中的最大元素并和 A、B 同样尺寸的数组。

C = max (A,[],dim)返回A中沿着由标量dim指定的维数上的最大元素。例如C=max(A,[],1)返回 A 中第一行上的最大元素。

[C, I] = max(...)寻找 A 中最大值的索引，并将其返回到输出向量 I 中。如果找到不止一个最大元素，则返回第一个的索引。

举例：

例如对于如下所示的一个任意矩阵：

```
A =
     1     2     3
     4     5     6
     0     2     3
    11    34     6
    67     1    34
```

利用函数 C = max (A)寻找各列中的最大元素，可得：

C = 67　34　34。

对于任意一个向量 A1 = 3　4　5　6　1　2　3　4　5，利用函数 C1 = max (A1)可以寻找 A1 中的最大元素，可得：C1 = 6。

再输入另一矩阵：

```
B =
    15    21    32
    40    51     6
    10    12    23
    18     3    16
    27    11    36
```

利用 C = max (A, B)寻找矩阵 A、B 中的最大元素，可得：

```
C =
    15    21    32
    40    51     6
    10    12    23
    18    34    16
    67    11    36
```

利用函数[C, I] = max(B)寻找 B 中最大值的索引，可得：

C = 40　51　36 和 I = 2　2　5。

10. 求数组的平均值

名称：mean

求数组的平均值。

语法：该函数有如下两种表达形式：

- M = mean (A)
- M = mean (A, dim)上的元素

描述：M = mean (A)返回 A 的平均值。

如果 A 是一个向量，则 mean (A)返回 A 中元素的平均值。

如果 A 是一个矩阵，则 mean (A)首先将 A 中的各列处理成向量，然后返回一个包含各列向量元素平均值的行向量。

M = mean (A, dim)返回 A 中沿着由标量 dim 指定的维数上的元素的平均值。

举例：

例如对于如下所示的一个任意矩阵：

```
A =
     1     2     3
     4     5     6
     0     2     3
    11    34     6
    67     1    34
```

利用函数 M = mean (A)求各列中元素的平均值，可得：

```
M =   16.6000    8.8000   10.4000
```

对于任意一个向量 A1 = 3 4 5 6 1 2 3 4 5，利用函数 M1= mean (A1)可以求得 A1 中元素的平均值为 M1 = 3.6667。

例如 M=mean(A, 2)返回 A 中各行元素的平均值：

```
M =
    2.0000
    5.0000
    1.6667
   17.0000
   34.0000
```

11. 求数组的中间值

名称：median

求数组的中间值。

语法：该函数有如下两种表达形式：

- M = median (A)
- M = median (A, dim)

描述：M = median (A)返回 A 的中间值。

如果 A 是一个向量，则 median (A)返回 A 中元素的中间值。

如果 A 是一个矩阵，则 median (A)首先将 A 中的各列处理成向量，然后返回一个包含

各列向量元素中间值的行向量。

M = median (A, dim)返回 A 中沿着由标量 dim 指定的维数上的元素的中间值。

举例：

例如对于如下所示的一个任意矩阵：

```
A =
      1      2      4      4
      3      4      6      6
      5      6      8      8
      5      6      8      8
```

利用函数 M = median (A)求各列中元素的中间值，可得：

```
M =   4      5      7      7
```

对于任意一个向量 A1 = 3 4 5 6 1 2 3 4 5，利用函数 M = median (A1)可以求得 A1 中元素的中间值为 M = 4。

例如 M= median (A, 2)返回 A 中各行元素的中间值：

```
M =
      3
      5
      7
      7
```

12. 求数组的最小元素

名称： min

求数组的最小元素。

语法： 该函数有如下几种表达形式：

- C = min (A)
- C = min (A, B)
- C = min (A,[],dim)
- [C,I] = min (...)

描述： C = min (A)返回 A 的最小元素。

如果 A 是一个向量，则 min (A)返回 A 中的最小元素。

如果 A 是一个矩阵，则 min (A)首先将 A 中的各列处理成向量，然后返回一个包含各列向量中的最小元素的行向量。

C = min (A, B)返回一个由 A、B 中最小元素组成的、并与 A、B 维数相同的数组。

C = min (A,[],dim)返回 A 中沿着由标量 dim 指定的维数上的最小元素。

[C, I] = min (...)寻找 A 中最小元素的索引，并将其返回到输出向量 I 中。如果有不止一个最小元素，则返回第一个的索引。

举例：

例如对于如下所示的一个任意矩阵：

```
A =
      1      2      4      4
```

3　4　6　6
5　6　8　8
5　6　8　8

利用函数 C = min (A)求各列中最小元素，可得：

C =　1　2　4　4

对于任意一个向量 A1 = 3　4　5　6　1　2　3　4　5，利用函数 C = min (A1)可以求得 A1 中最小元素为 C = 1。

再取另一个矩阵：

B =

3　5　6　1
8　2　5　1
5　8　0　3
0　4　7　1

利用函数 C = min (A, B)可以求得一个由 A、B 中最小元素组成的、并与 A、B 维数相同的数组 C：

C =

1　2　4　1
3　2　5　1
5　6　0　3
0　4　7　1

函数 C = min (A,[],2)返回 A 中各行的最小元素：

C =

1
3
5
5

13. 所有可能变换

名称：perms

所有可能变换。

语法：P = perms (v)

描述：P = perms (v)返回一个包含有 v 中 n 个元素的所有可能变换的矩阵，其中 v 是一个长度为 n 的行向量。

举例：

例如对于向量 v = 2　4　6，函数 P = perms (v)返回矩阵：

P =

6　4　2
4　6　2
6　2　4
2　6　4

```
    4    2    6
    2    4    6
```

14. 多边形的面积

名称：polyarea

多边形的面积。

语法：该函数有如下两种表达形式：

- A = polyarea (X, Y)
- A = polyarea (X, Y, dim)

描述：A = polyarea (X, Y)返回以向量 X 和 Y 为顶点坐标所组成的多边形的面积。

如果 X 和 Y 是维数相同的矩阵，则函数 polyarea 返回由 X 和 Y 中列向量所确定的多边形的面积。

A = polyarea (X, Y, dim)沿着由标量 dim 指定的维数进行上述操作。

举例：

本例中首先绘制并显示多边形，然后计算该多边形的面积。

首先在 MATLAB 命令窗口中输入 L = linspace (0, 2.*pi, 6)确定坐标线，得到：

```
L = 0    1.2566    2.5133    3.7699    5.0265    6.2832
```

然后给定多边形各顶点的坐标，输入 xv = cos(L)'，得到对应的 x 坐标：

```
xv =
    1.0000
    0.3090
   -0.8090
   -0.8090
    0.3090
    1.0000
```

同样由 yv = sin(L)'得到 y 坐标：

```
yv =
         0
    0.9511
    0.5878
   -0.5878
   -0.9511
   -0.0000
```

利用函数 A = polyarea(xv,yv)计算多边形的面积为 A = 2.3776。

为了将该多边形在窗口中进行显示，执行 plot(xv,yv)命令，并利用函数 title(['Area = ' num2str(A)])为该图形加上标题，如图 7-4 所示。

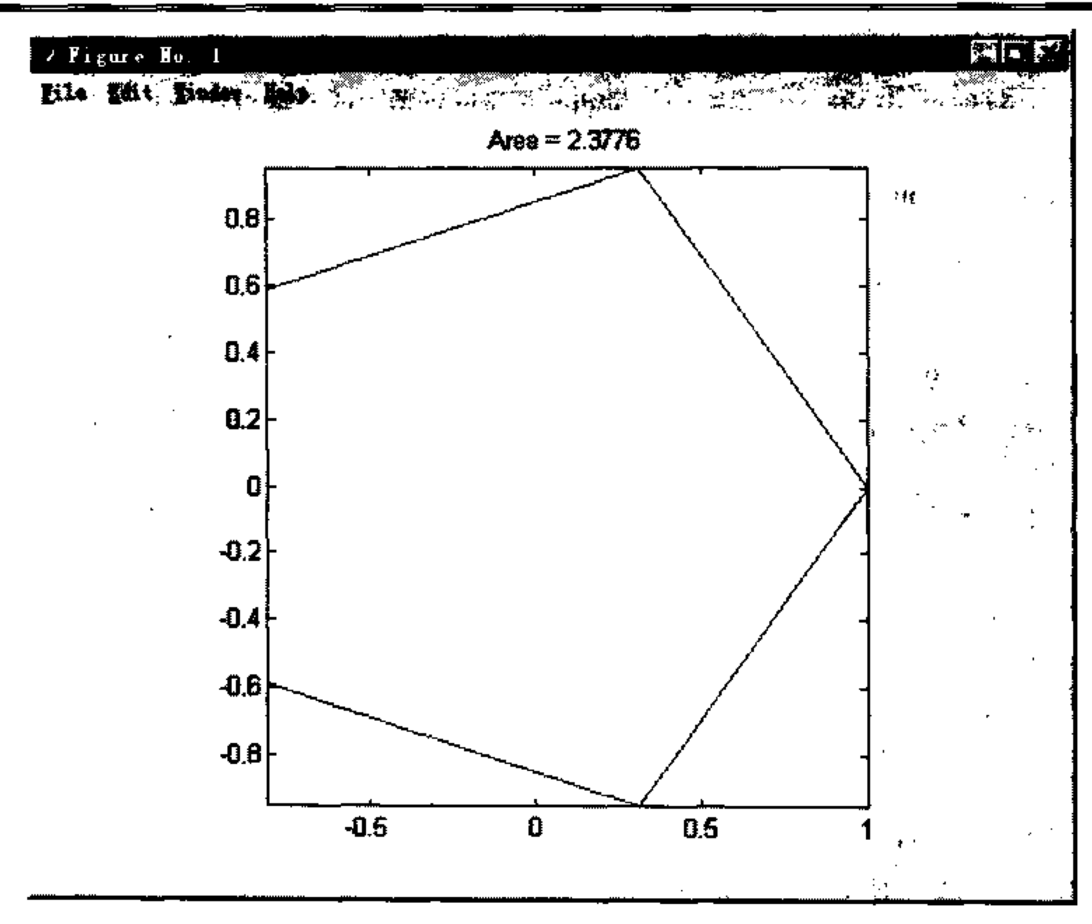


图 7-4　多边形显示窗口

15. 生成质数列表

名称：primes

生成质数列表。

语法：p = primes (n)

描述：p = primes (n)返回一个由所有小于或等于 n 的质数所组成的行向量。

所谓质数就是除了 1 和本身之外没有其他分解因子的数。

举例：

例如取 n = 37，利用函数 p = primes (n)可以得到一个由所有小于或等于 37 的质数所组成的行向量 p：

p = 2　3　5　7　11　13　17　19　23　29　31　37。

16. 数组元素积

名称：prod

求解数组元素积。

语法：该函数有如下两种表达形式：

- B = prod (A)
- B = prod (A, dim)

描述：B = prod (A)返回数组元素的积。

如果 A 是一个向量，则 prod (A)返回 A 中各个元素的积。

如果 A 是一个矩阵，则 prod (A)首先将 A 中的各列处理成向量，然后返回一个包含各

列向量元素积的行向量。

B = prod (A, dim)返回 A 中沿着由标量 dim 指定的维数上的元素积。

举例：

例如对于 3 阶 magic 矩阵：

M =

```
     8     1     6
     3     5     7
     4     9     2
```

利用函数 B = prod (M)可以得到由矩阵 A 中各列向量元素的积所组成的一个行向量 B：

B = 96　　45　　84。

函数 B = prod (M, 2)可以得到由矩阵 A 中各行向量元素的积所组成的一个列向量 B：

B =

```
    48
   105
    72
```

17. 将元素按升序排列

名称： sort

将元素按升序排列。

语法： 该函数有如下几种表达形式：

- B = sort (A)
- [B, INDEX] = sort (A)
- B = sort (A, dim)

描述： B = sort (A)将数组 A 中的元素按升序进行排列。其中 A 中的元素可以是实数或复数，也可以是字符串。

如果 A 是一个向量，则 sort (A)将 A 中各个元素按升序排列。

如果 A 是一个矩阵，则 sort (A)首先将 A 中的各列处理成向量，然后返回经过重新升序排列后的列向量。

[B, INDEX] = sort (A)还返回一个索引数组 INDEX。该数组等于 size(A)，它的每一列是 A 数组中相应列的置换向量。

B = sort (A, dim)对 A 中沿着由标量 dim 指定的元素进行重新排序。

如果 dim 是一个向量，则函数 sort 对特定的维数上的元素进行迭代处理。例如 sort(A,[1 2])就等效于 sort(sort(A,2),1)。

举例：

例如对于一个任意矩阵

A =

```
     3     5     9
     2     5     6
    14     9     7
```

利用函数 B = sort (A)可以对矩阵 A 中各个列向量的元素进行升序排列，结果如下所示：

B =

2	5	6
3	5	7
14	9	9

为了得到索引数组 INDEX，在 MATLAB 命令窗口中输入：[B, INDEX] = sort (A)，可以得到：

B =

2	5	6
3	5	7
14	9	9

INDEX =

2	1	2
1	2	3
3	3	1

为了对矩阵 A 中各个行向量的元素进行升序排列，可以利用函数 B = sort (A, 2)，结果如下所示：

B =

3	5	9
2	5	6
7	9	14

18. 将行按升序排列

名称：sortrows

将行按升序排列。

语法：该函数有如下几种表达形式：

- B = sortrows (A)
- B = sortrows (A, column)
- [B, index] = sortrows (A)

描述：B = sortrows (A)将 A 中的所有行作为一组元素，并对其按升序进行排列。A 必须是矩阵或列向量。

B = sortrows (A, column)基于向量 column 指定的列对矩阵 A 按升序进行排列。例如 sortrows(A,[2 3])将基于第 2 列对矩阵 A 进行重排序。

[B, index] = sortrows (A)还将返回一个索引向量 index。

如果 A 是一个列向量，则 B = A(index)。

如果 A 是一个 m×n 矩阵，则 B = A(index,:)。

举例：

例如对于一个给定的 5×5 (字符串)矩阵

A =

one

two

three
four
five

利用函数 B = sortrows (A)可以对 A 的行进行重排(按字母升序排列)，结果如下所示：

B =
five
four
one
three
two

如果想基于第 2 列中的元素对矩阵 A 进行重排，可以使用函数 B = sortrows (A, 2)，得到：

B =
three
five
one
four
two

函数[B, index] = sortrows (A)将同时返回一个索引向量 index：

B =
five
four
one
three
two
index =
    5
    4
    1
    3
    2

19. 标准差

名称：std

标准差。

语法：该函数有如下几种表达形式：

- s = std (X)
- s = std (X, flag)
- s = std (X, flag, dim)

描述：在现行的教材中，有两种定义向量 x 标准差的公式：

$$s=\left(\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{1/2} \text{和} s=\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{1/2}。$$

其中 $\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i$，n 为样本中元素的数目。两个公式的不同之处在于分母中 n-1 和 n 的差别。

若 X 是一个向量，则 std (X)返回采用公式(1)计算得到的标准差。

若 X 是一个矩阵，则 std (X)返回一个包含 X 中每一列元素标准差的行向量。

当 flag=0 时，s = std (X, flag)与 s = std (X)等价。

当 flag=1 时，s = std (X, flag)返回采用公式(2)计算得到的标准差。

s = std (X, flag, dim)将按标量 dim 指定的维数计算 X 中元素的标准差。

举例：

例如对于如下所示的矩阵：

```
X =
     1     5     9
     7    25    22
```

利用函数 s = std (X)可以计算(利用公式(1))X 中元素的标准差，结果为：

s = 4.2426　　14.1421　　9.1924。

若想利用公式(2)计算标准差，可执行函数 s = std (X, 1)，结果为：

s = 3.0000　　10.0000　　6.5000。

函数 s = std (X, 0, 2)可以得到按行计算得到的标准差，如下所示：

```
s =
    4.0000
    9.6437
```

20. 数组元素的和

名称：sum

数组元素的和。

语法：该函数有如下两种表达形式：

- B = sum (A)
- B = sum (A, dim)

描述：B = sum (A)返回数组中元素的和。

如果 A 是一个向量，则 sum (A)返回 A 中各个元素的和。

如果 A 是一个矩阵，则 sum (A) 首先将 A 中的各列处理成向量，然后返回一个包含各个列向量元素和的行向量。

B = sum (A, dim) 将按标量 dim 指定的维数计算 A 中相应元素的和。

举例：

例如对于如下所示的 magic 矩阵：

A =

```
8   1   6
3   5   7
4   9   2
```

函数 B = sum (A)将返回一个包含 A 中各个列向量中元素和的行向量，结果如下所示：

B = 15   15   15。

若想得到 A 中各个行向量中元素的和，可执行函数 B = sum (A, 2)，结果为：

```
B =
    15
    15
    15
```

21. 梯形数值积分

名称：trapz

梯形数值积分。

语法：该函数有如下几种表达形式：

- Z = trapz (Y)
- Z = trapz (X, Y)
- Z = trapz (..., dim)

描述：Z = trapz (Y)采用梯形法计算 Y 的近似积分。

如果 Y 是一个向量，则 trapz (Y)就是 Y 的积分。

如果 Y 是一个矩阵，则 trapz (Y)是一个包含 Y 中各列元素积分的行向量。

Z = trapz (X, Y)返回采用梯形法并基于 X 计算得到的 Y 的积分。

Z = trapz (..., dim)沿着标量 dim 指定的维数对 Y 进行积分。

举例：

积分 $\int_0^{\pi}\sin(x)\mathrm{d}x$ 的精确值为 2。

为了得到该积分的近似数值积分解，令：

X = 0:pi/100:pi，即：X =

```
Columns 1 through 7
         0    0.0314    0.0628    0.0942    0.1257    0.1571    0.1885
  Columns 8 through 14
    0.2199    0.2513    0.2827    0.3142    0.3456    0.3770    0.4084
  Columns 15 through 21
    0.4398    0.4712    0.5027    0.5341    0.5655    0.5969    0.6283
  Columns 22 through 28
    0.6597    0.6912    0.7226    0.7540    0.7854    0.8168    0.8482
  Columns 29 through 35
    0.8796    0.9111    0.9425    0.9739    1.0053    1.0367    1.0681
  Columns 36 through 42
    1.0996    1.1310    1.1624    1.1938    1.2252    1.2566    1.2881
```

```
  Columns 43 through 49
    1.3195    1.3509    1.3823    1.4137    1.4451    1.4765    1.5080
  Columns 50 through 56
    1.5394    1.5708    1.6022    1.6336    1.6650    1.6965    1.7279
  Columns 57 through 63
    1.7593    1.7907    1.8221    1.8535    1.8850    1.9164    1.9478
  Columns 64 through 70
    1.9792    2.0106    2.0420    2.0735    2.1049    2.1363    2.1677
  Columns 71 through 77
    2.1991    2.2305    2.2619    2.2934    2.3248    2.3562    2.3876
  Columns 78 through 84
    2.4190    2.4504    2.4819    2.5133    2.5447    2.5761    2.6075
  Columns 85 through 91
    2.6389    2.6704    2.7018    2.7332    2.7646    2.7960    2.8274
  Columns 92 through 98
    2.8588    2.8903    2.9217    2.9531    2.9845    3.0159    3.0473
  Columns 99 through 101
    3.0788    3.1102    3.1416
```

令 Y = sin(X)，即 Y =

```
Columns 1 through 7
         0    0.0314    0.0628    0.0941    0.1253    0.1564    0.1874
  Columns 8 through 14
    0.2181    0.2487    0.2790    0.3090    0.3387    0.3681    0.3971
  Columns 15 through 21
    0.4258    0.4540    0.4818    0.5090    0.5358    0.5621    0.5878
  Columns 22 through 28
    0.6129    0.6374    0.6613    0.6845    0.7071    0.7290    0.7501
  Columns 29 through 35
    0.7705    0.7902    0.8090    0.8271    0.8443    0.8607    0.8763
  Columns 36 through 42
    0.8910    0.9048    0.9178    0.9298    0.9409    0.9511    0.9603
  Columns 43 through 49
    0.9686    0.9759    0.9823    0.9877    0.9921    0.9956    0.9980
  Columns 50 through 56
    0.9995    1.0000    0.9995    0.9980    0.9956    0.9921    0.9877
  Columns 57 through 63
    0.9823    0.9759    0.9686    0.9603    0.9511    0.9409    0.9298
  Columns 64 through 70
    0.9178    0.9048    0.8910    0.8763    0.8607    0.8443    0.8271
```

```
  Columns 71 through 77
    0.8090    0.7902    0.7705    0.7501    0.7290    0.7071    0.6845
  Columns 78 through 84
    0.6613    0.6374    0.6129    0.5878    0.5621    0.5358    0.5090
  Columns 85 through 91
    0.4818    0.4540    0.4258    0.3971    0.3681    0.3387    0.3090
  Columns 92 through 98
    0.2790    0.2487    0.2181    0.1874    0.1564    0.1253    0.0941
  Columns 99 through 101
    0.0628    0.0314    0.0000
```

利用函数 Z = trapz(X,Y)计算上述积分，可得：Z = 1.9998。

22. 搜索 Delaunay 三角形

名称：tsearch

搜索 Delaunay 三角形。

语法：T = tsearch (x, y, TRI, xi, yi)

描述：T = tsearch (x, y, TRI, xi, yi)返回向量 xi 和 yi 所包含的各点中属于 Delaunay 三角形的所有点的索引。对于所有位于凸壳外部的点，tsearch 函数将返回 NaN。

举例：

首先对一组给定点进行 Delaunay 三角分解。

利用函数 x = rand(1,10)和 y = rand(1,10)分别产生两组随机数：

```
x =
  Columns 1 through 7
    0.3400    0.3142    0.3651    0.3932    0.5915    0.1197    0.0381
  Columns 8 through 10
    0.4586    0.8699    0.9342
y =
  Columns 1 through 7
    0.2644    0.1603    0.8729    0.2379    0.6458    0.9669    0.6649
  Columns 8 through 10
    0.8704    0.0099    0.1370
```

利用函数 TRI = delaunay (x, y)进行 Delaunay 三角分解，可得：

```
TRI =
     1     4     2
     4     9     2
     4    10     9
     5    10     4
     2     7     1
     1     5     4
     7     5     1
```

3	8	5
7	3	5
6	3	7
6	8	3
8	10	5

利用函数 xi = rand(1,5)和 yi = rand(1,5)再产生两组随机数:

xi = 0.8188 0.4302 0.8903 0.7349 0.6873

yi = 0.3461 0.1660 0.1556 0.1911 0.4225。

为了搜索 Delaunay 三角形，在 MATLAB 命令窗口中输入 T = tsearch (x, y, TRI, xi, yi)，可得：T = NaN 2 4 4 4。

23. 方差

名称：var

方差。

语法：该函数有如下几种表达形式：

- var (X)
- var (X, 1)
- var (X, w)

描述：当 X 是向量时，var (X)返回 X 的方差。

当 X 是矩阵时，var (X)返回一个包含 X 中各列元素方差的行向量。

var (X, w)使用权向量 w 计算矩阵 X 的方差。其中，w 中的元素数要等于(除了 w=1)矩阵 X 中的行数，而且 w 中的元素必须是正数。函数 var 通过将矩阵 X 中的每一个元素除以所有元素的和来规格化 X。

举例：

例如对于矩阵

X =

2	3	6
1	5	7
9	4	6

利用函数 var (X)可以计算其方差为：19.0000 1.0000 0.3333。

利用函数 var (X, [1 2 3])可以使用全向量[1 2 3]来计算 X 的方差，结果为：

14.8056 0.4722 0.2222。

24. Voronoi 图

名称：voronoi

Voronoi 图。

语法：该函数有如下几种表达形式：

- voronoi (x, y)
- voronoi (x, y, TRI)
- h = voronoi (..., 'LineSpec')
- [vx, vy] = voronoi(...)

描述：考虑一个给定的共面点集 P。对于该点集中的任何一点 P_x，可以绘制一个包含该点的、而且距离该点最近的边界，该边界被称为 Voronoi 多边形。该共面点集的所有 Voronoi 多边形的集合称为 Voronoi 图。

voronoi (x, y)绘制由向量 x 和 y 所描述的点的 Voronoi 图。

voronoi (x, y, TRI)使用三角形 TRI，而不是通过 delaunay 来计算。

h = voronoi (..., 'LineSpec')用指定的颜色和线型绘制图形，并返回 h 中创建的线性对象的句柄。

[vx, vy] = voronoi(...)在向量 vx 和 vy 中返回 Voronoi 边界的顶点，从而可以利用函数 plot(vx,vy,'-',x,y,'.')来绘制 Voronoi 图。

举例：

下面的例子对 10 个随机产生的点绘制了其 Voronoi 图。

首先在 MATLAB 命令窗口中输入 rand('state',0)设置随机数产生的方式。接着用函数 x = rand(1,10)和 y = rand(1,10)分别产生两组随机数，如下所示：

```
x =
  Columns 1 through 7
    0.9501    0.2311    0.6068    0.4860    0.8913    0.7621    0.4565
  Columns 8 through 10
    0.0185    0.8214    0.4447
y =
  Columns 1 through 7
    0.6154    0.7919    0.9218    0.7382    0.1763    0.4057    0.9355
  Columns 8 through 10
    0.9169    0.4103    0.8936
```

然后利用函数[vx, vy] = voronoi(x,y)绘制由向量 x 和 y 所描述的点的 Voronoi 图。结果如下所示：

```
vx =
  Columns 1 through 7
    0.7090    0.3636    0.7271   -0.1117    0.7785    0.2529    0.7090
    0.7271    0.2529    0.5384   -1.2073    0.7090   -0.1117    0.2529
  Columns 8 through 14
   -0.1117    0.2327   -1.2073    1.1134    0.8018    0.7785    0.2810
    0.8018    0.2810    0.2327    0.7785    1.1134    0.8018    0.3636
  Columns 15 through 17
    0.5384    0.3636    0.5284
    0.5284    0.5384    0.2810
vy =
  Columns 1 through 7
    0.6425    0.7889    0.7111   -0.2374    0.5802    0.2637    0.6425
    0.7111    0.2637    0.8353   -1.4122    0.6425   -0.2374    0.2637
```

Columns 8 through 14

-0.2374	1.0381	-1.4122	0.3700	0.2770	0.5802	0.9623
0.2770	0.9623	1.0381	0.5802	0.3700	0.2770	0.7889

Columns 15 through 17

0.8353	0.7889	0.8927
0.8927	0.8353	0.9623

最后利用函数 plot(x,y,'r+',vx,vy,'b-')和 axis equal 在窗口中显示所绘制的 Voronoi 图，如图 7-5 所示。

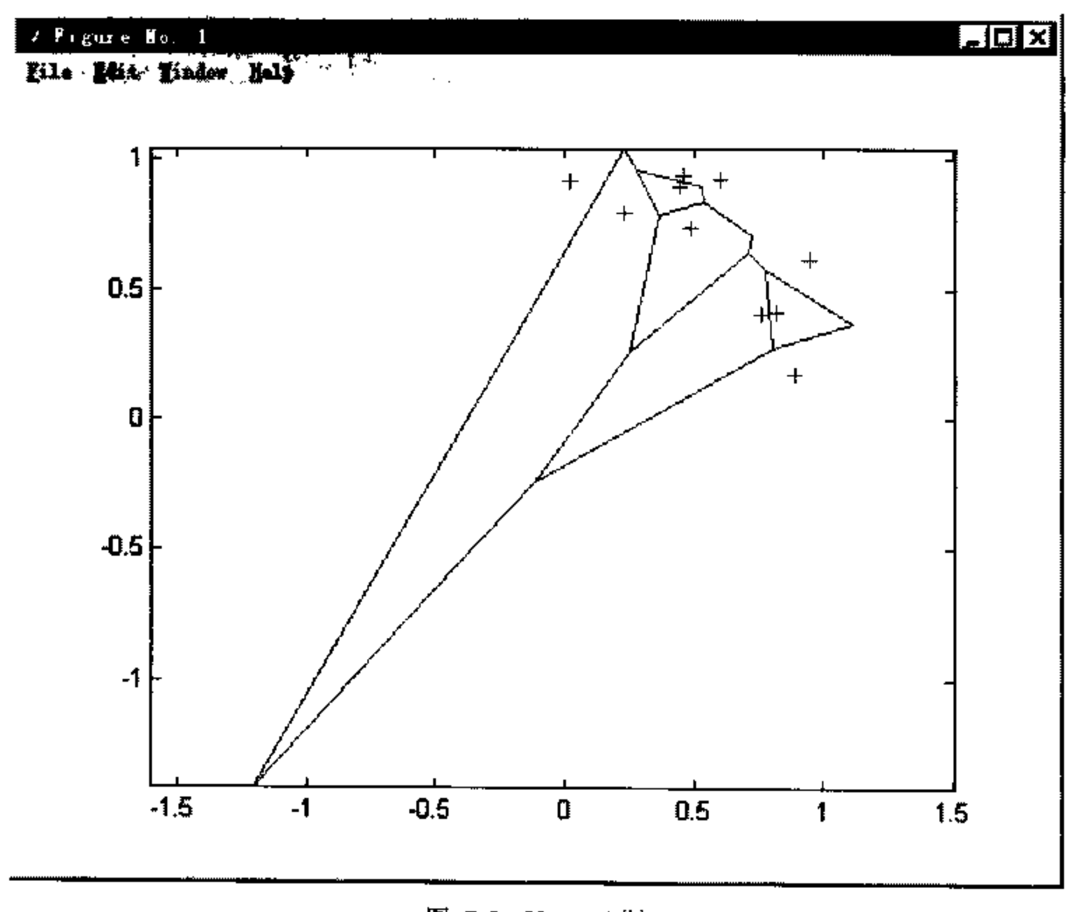


图 7-5　Voronoi 图

7.2　有限差分

1. Laplacian 离散

名称：del2

Laplacian 离散。

语法：该函数有如下几种表达形式：

- L = del2 (U)
- L = del2 (U, h)
- L = del2 (U, hx, hy)

- L = del2 (U, hx, hy, hz,...)

描述: 如果把矩阵 U 看作是函数 $u(x, y)$ 在正方形网格上某一点处的取值，那么 4*del2(U) 就是将 Laplace 微分算子作用于 u 的有限差分近似，即：

$l = \dfrac{\nabla^2 u}{4} = \dfrac{1}{4}\left(\dfrac{d^2 u}{dx^2} + \dfrac{d^2 u}{dy^2}\right)$，其中在内部 $l_{ij} = \dfrac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) - u_{i,j}$。

对于包含更多变量的函数 $u(x, y, z, \ldots)$，del2(U)是下式的近似：

$l = \dfrac{\nabla^2 u}{2N} = \dfrac{1}{2N}\left(\dfrac{d^2 u}{dx^2} + \dfrac{d^2 u}{dy^2} + \dfrac{d^2 u}{dz^2} + \ldots\right)$，其中 N 是函数 $u(x, y, z, \ldots)$ 中的变量数。

当 U 是一个矩形数组时，L = del2 (U)为 $l = \dfrac{\nabla^2 u}{4} = \dfrac{1}{4}\left(\dfrac{d^2 u}{dx^2} + \dfrac{d^2 u}{dy^2}\right)$ 的离散近似。而且矩阵 L 和 U 具有相同的维数。

当 U 是一个多维数组时，L = del2 (U)返回 $\dfrac{\nabla^2 u}{2N}$ 的近似，其中 N= ndims(u)。

L = del2 (U, h)中 h 为一个标量，它表示各个方向上相邻点之间的间距，缺省值为 1。

当 U 是一个矩形数组时，L = del2 (U, hx, hy)利用 hx 和 hy 的值作为间距进行离散。

如果 hx 是一个标量，则给定 x 方向上相邻点之间的间距。

如果 hx 是一个向量，则等于 size(u,2)，并给定了点的 x 坐标。

类似地，如果 hy 是一个标量，则给定 y 方向上相邻点之间的间距。

如果 hy 是一个向量，则等于 size(u,1)，并给定了点的 y 坐标。

L = del2 (U, hx, hy, hz,...)中 U 是多维数组，hx、hy、hz...为各个方向上的间距。

举例:

例如对于函数 $u(x, y) = x^2 + y^2$，$\nabla^2 u = 4$。

在 MATLAB 命令窗口中输入[x,y] = meshgrid(-4:4,-3:3)，指定 x 和 y 数组的取值：

```
x =
    -4    -3    -2    -1     0     1     2     3     4
    -4    -3    -2    -1     0     1     2     3     4
    -4    -3    -2    -1     0     1     2     3     4
    -4    -3    -2    -1     0     1     2     3     4
    -4    -3    -2    -1     0     1     2     3     4
    -4    -3    -2    -1     0     1     2     3     4
    -4    -3    -2    -1     0     1     2     3     4
y =
    -3    -3    -3    -3    -3    -3    -3    -3    -3
    -2    -2    -2    -2    -2    -2    -2    -2    -2
    -1    -1    -1    -1    -1    -1    -1    -1    -1
     0     0     0     0     0     0     0     0     0
     1     1     1     1     1     1     1     1     1
     2     2     2     2     2     2     2     2     2
```

```
     3     3     3     3     3     3     3     3     3
```

利用函数 U = x.*x+y.*y 确定矩阵 U 的元素:

```
U =
    25    18    13    10     9    10    13    18    25
    20    13     8     5     4     5     8    13    20
    17    10     5     2     1     2     5    10    17
    16     9     4     1     0     1     4     9    16
    17    10     5     2     1     2     5    10    17
    20    13     8     5     4     5     8    13    20
    25    18    13    10     9    10    13    18    25
```

最后利用函数 V = 4*del2(U)进行离散，可得:

```
V =
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
```

若用函数 L = del2 (U, 4)进行离散，可得:

```
L =
  Columns 1 through 7
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625    0.0625
  Columns 8 through 9
    0.0625    0.0625
    0.0625    0.0625
    0.0625    0.0625
    0.0625    0.0625
    0.0625    0.0625
    0.0625    0.0625
    0.0625    0.0625
```

2. 差分和近似微分

名称：diff

差分和近似微分。

语法：该函数有如下几种表达形式：

- Y = diff (X)
- Y = diff (X, n)
- Y = diff (X, n, dim)

描述：Y = diff (X)返回 X 中相邻元素之间的差。

如果 X 是一个向量，则 diff (X)返回一个包含相邻元素之间的差值、并比 X 的维数少 1 的向量：$[X(2)-X(1) \quad X(3)-X(2) \quad \dots \quad X(n)-X(n-1)]$。

如果 X 是一个矩阵，则 diff (X) 返回一个包含列向量差值的矩阵：$[X(2:m,:)-X(1:m-1,:)]$。

Y = diff (X, n)反复调用 diff 函数 n 次，最后得到第 n 阶微分。例如函数 diff(X,2)与函数 diff(diff(X))等价。

Y = diff (X, n, dim)是沿着由标量 dim 指定的维数计算得到的第 n 阶微分函数。

举例：

对于向量 X = 1 2 3 4 5，执行函数 y = diff(x)可以得到 X 中相邻元素之间的差：y = 1 1 1 1。

利用函数 Y = diff (X, 2)可以得到 X 的 2 阶微分：Y = 0 0 0。

对于给定的数组 A = rand(1,3,2,4)，即：

```
A(:,:,1,1) =
     0.0579     0.3529     0.8132
A(:,:,2,1) =
     0.0099     0.1389     0.2028
A(:,:,1,2) =
     0.1987     0.6038     0.2722
A(:,:,2,2) =
     0.1988     0.0153     0.7468
A(:,:,1,3) =
     0.4451     0.9318     0.4660
A(:,:,2,3) =
     0.4186     0.8462     0.5252
A(:,:,1,4) =
     0.2026     0.6721     0.8381
A(:,:,2,4) =
     0.0196     0.6813     0.3795
```

函数 diff(A)可以得到 A 沿着维数 2 的 1 阶微分，如下所示：

```
ans(:,:,1,1) =
     0.2950     0.4603
ans(:,:,2,1) =
```

```
     0.1290      0.0639
ans(:,:,1,2) =
     0.4051    -0.3316
ans(:,:,2,2) =
    -0.1835     0.7315
ans(:,:,1,3) =
     0.4867    -0.4658
ans(:,:,2,3) =
     0.4276    -0.3211
ans(:,:,1,4) =
     0.4695     0.1660
ans(:,:,2,4) =
     0.6616    -0.3018
```

利用函数 diff(A,3,4)可以得到 A 沿着维数 4 的 3 阶微分，如下所示：

```
ans(:,:,1) =
    -0.5944    -0.6648    -0.5565
ans(:,:,2) =
    -0.6497    -1.9505     0.8416
```

3. 数值梯度

名称：gradient

数值梯度。

语法：该函数有如下几种表达形式：

- FX = gradient(F)
- [FX,FY] = gradient(F)
- [Fx,Fy,Fz,...] = gradient(F)
- [...] = gradient(F,h)
- [...] = gradient(F,h1,h2,...)

描述：包含两个变量的函数 $F(x,y)$ 的梯度可用下式来定义：

$\nabla F = \frac{\partial F}{\partial x}\hat{\mathbf{i}} + \frac{\partial F}{\partial y}\hat{\mathbf{j}}$。该梯度可以看作是指向使 F 的取值增加的方向的向量集。

用 MATLAB 能够计算包含任意个变量的函数的数值梯度。

FX = gradient(F)返回 F 的一维数值梯度。其中 F 是一个向量，FX 相应于 $\partial F/\partial x$ (即在 x 方向上的微分)。

[FX,FY] = gradient(F)返回 F 的数值梯度在 x 和 y 方向上的分量。其中 F 是一个矩阵，FX 相应于 $\partial F/\partial x$ (即在 x 方向上的微分)；FY 相应于 $\partial F/\partial y$ (即在 y 方向上的微分)。

[Fx,Fy,Fz,...] = gradient(F)返回 F 的数值梯度的 n 个分量。其中 F 是一个 n 维数组。

[...] = gradient(F,h)中的 h 是一个标量，表示各个方向上相邻点之间的间距。

[...] = gradient(F,h1,h2,...)利用 N 个间距参数来分别指定 N 个方向上相邻点之间的间距。

举例:

首先利用函数 v = -2:0.2:2 得到一个一维数组:

```
v =
  Columns 1 through 7
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
  Columns 8 through 14
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
  Columns 15 through 21
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
```

然后利用函数[x,y] = meshgrid(v)分别确定相应的 x 数组和 y 数组:

```
x =
  Columns 1 through 7
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
   -2.0000   -1.8000   -1.6000   -1.4000   -1.2000   -1.0000   -0.8000
  Columns 8 through 14
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
```

```
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
   -0.6000   -0.4000   -0.2000         0    0.2000    0.4000    0.6000
  Columns 15 through 21
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
y =
  Columns 1 through 7
   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000
   -1.8000   -1.8000   -1.8000   -1.8000   -1.8000   -1.8000   -1.8000
   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000
   -1.4000   -1.4000   -1.4000   -1.4000   -1.4000   -1.4000   -1.4000
   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000
   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000
   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000
   -0.6000   -0.6000   -0.6000   -0.6000   -0.6000   -0.6000   -0.6000
   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000
   -0.2000   -0.2000   -0.2000   -0.2000   -0.2000   -0.2000   -0.2000
         0         0         0         0         0         0         0
    0.2000    0.2000    0.2000    0.2000    0.2000    0.2000    0.2000
    0.4000    0.4000    0.4000    0.4000    0.4000    0.4000    0.4000
    0.6000    0.6000    0.6000    0.6000    0.6000    0.6000    0.6000
    0.8000    0.8000    0.8000    0.8000    0.8000    0.8000    0.8000
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
    1.2000    1.2000    1.2000    1.2000    1.2000    1.2000    1.2000
    1.4000    1.4000    1.4000    1.4000    1.4000    1.4000    1.4000
    1.6000    1.6000    1.6000    1.6000    1.6000    1.6000    1.6000
    1.8000    1.8000    1.8000    1.8000    1.8000    1.8000    1.8000
    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000
```

```
  Columns 8 through 14
   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000
   -1.8000   -1.8000   -1.8000   -1.8000   -1.8000   -1.8000   -1.8000
   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000
   -1.4000   -1.4000   -1.4000   -1.4000   -1.4000   -1.4000   -1.4000
   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000
   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000
   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000
   -0.6000   -0.6000   -0.6000   -0.6000   -0.6000   -0.6000   -0.6000
   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000
   -0.2000   -0.2000   -0.2000   -0.2000   -0.2000   -0.2000   -0.2000
         0         0         0         0         0         0         0
    0.2000    0.2000    0.2000    0.2000    0.2000    0.2000    0.2000
    0.4000    0.4000    0.4000    0.4000    0.4000    0.4000    0.4000
    0.6000    0.6000    0.6000    0.6000    0.6000    0.6000    0.6000
    0.8000    0.8000    0.8000    0.8000    0.8000    0.8000    0.8000
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
    1.2000    1.2000    1.2000    1.2000    1.2000    1.2000    1.2000
    1.4000    1.4000    1.4000    1.4000    1.4000    1.4000    1.4000
    1.6000    1.6000    1.6000    1.6000    1.6000    1.6000    1.6000
    1.8000    1.8000    1.8000    1.8000    1.8000    1.8000    1.8000
    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000
  Columns 15 through 21
   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000
   -1.8000   -1.8000   -1.8000   -1.8000   -1.8000   -1.8000   -1.8000
   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000
   -1.4000   -1.4000   -1.4000   -1.4000   -1.4000   -1.4000   -1.4000
   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000
   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000
   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000
   -0.6000   -0.6000   -0.6000   -0.6000   -0.6000   -0.6000   -0.6000
   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000
   -0.2000   -0.2000   -0.2000   -0.2000   -0.2000   -0.2000   -0.2000
         0         0         0         0         0         0         0
    0.2000    0.2000    0.2000    0.2000    0.2000    0.2000    0.2000
    0.4000    0.4000    0.4000    0.4000    0.4000    0.4000    0.4000
    0.6000    0.6000    0.6000    0.6000    0.6000    0.6000    0.6000
    0.8000    0.8000    0.8000    0.8000    0.8000    0.8000    0.8000
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
    1.2000    1.2000    1.2000    1.2000    1.2000    1.2000    1.2000
    1.4000    1.4000    1.4000    1.4000    1.4000    1.4000    1.4000
    1.6000    1.6000    1.6000    1.6000    1.6000    1.6000    1.6000
    1.8000    1.8000    1.8000    1.8000    1.8000    1.8000    1.8000
    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000
```

利用函数 z = x .* exp(-x.^2 - y.^2)计算得到：

```
z =
  Columns 1 through 7
   -0.0007   -0.0013   -0.0023   -0.0036   -0.0052   -0.0067   -0.0077
   -0.0014   -0.0028   -0.0048   -0.0077   -0.0111   -0.0144   -0.0165
   -0.0028   -0.0054   -0.0096   -0.0152   -0.0220   -0.0284   -0.0326
   -0.0052   -0.0099   -0.0174   -0.0278   -0.0400   -0.0518   -0.0594
```

```
  -0.0087   -0.0167   -0.0293   -0.0467   -0.0674   -0.0872   -0.0999
  -0.0135   -0.0259   -0.0455   -0.0725   -0.1046   -0.1353   -0.1552
  -0.0193   -0.0372   -0.0652   -0.1040   -0.1499   -0.1940   -0.2224
  -0.0256   -0.0492   -0.0863   -0.1376   -0.1984   -0.2567   -0.2943
  -0.0312   -0.0601   -0.1054   -0.1680   -0.2423   -0.3135   -0.3595
  -0.0352   -0.0677   -0.1188   -0.1895   -0.2732   -0.3535   -0.4053
  -0.0366   -0.0705   -0.1237   -0.1972   -0.2843   -0.3679   -0.4218
  -0.0352   -0.0677   -0.1188   -0.1895   -0.2732   -0.3535   -0.4053
  -0.0312   -0.0601   -0.1054   -0.1680   -0.2423   -0.3135   -0.3595
  -0.0256   -0.0492   -0.0863   -0.1376   -0.1984   -0.2567   -0.2943
  -0.0193   -0.0372   -0.0652   -0.1040   -0.1499   -0.1940   -0.2224
  -0.0135   -0.0259   -0.0455   -0.0725   -0.1046   -0.1353   -0.1552
  -0.0087   -0.0167   -0.0293   -0.0467   -0.0674   -0.0872   -0.0999
  -0.0052   -0.0099   -0.0174   -0.0278   -0.0400   -0.0518   -0.0594
  -0.0028   -0.0054   -0.0096   -0.0152   -0.0220   -0.0284   -0.0326
  -0.0014   -0.0028   -0.0048   -0.0077   -0.0111   -0.0144   -0.0165
  -0.0007   -0.0013   -0.0023   -0.0036   -0.0052   -0.0067   -0.0077
 Columns 8 through 14
  -0.0077   -0.0062   -0.0035         0    0.0035    0.0062    0.0077
  -0.0164   -0.0133   -0.0075         0    0.0075    0.0133    0.0164
  -0.0324   -0.0263   -0.0149         0    0.0149    0.0263    0.0324
  -0.0590   -0.0480   -0.0271         0    0.0271    0.0480    0.0590
  -0.0992   -0.0808   -0.0455         0    0.0455    0.0808    0.0992
  -0.1540   -0.1254   -0.0707         0    0.0707    0.1254    0.1540
  -0.2207   -0.1797   -0.1013         0    0.1013    0.1797    0.2207
  -0.2921   -0.2378   -0.1341         0    0.1341    0.2378    0.2921
  -0.3567   -0.2905   -0.1637         0    0.1637    0.2905    0.3567
  -0.4022   -0.3275   -0.1846         0    0.1846    0.3275    0.4022
  -0.4186   -0.3409   -0.1922         0    0.1922    0.3409    0.4186
  -0.4022   -0.3275   -0.1846         0    0.1846    0.3275    0.4022
  -0.3567   -0.2905   -0.1637         0    0.1637    0.2905    0.3567
  -0.2921   -0.2378   -0.1341         0    0.1341    0.2378    0.2921
  -0.2207   -0.1797   -0.1013         0    0.1013    0.1797    0.2207
  -0.1540   -0.1254   -0.0707         0    0.0707    0.1254    0.1540
  -0.0992   -0.0808   -0.0455         0    0.0455    0.0808    0.0992
  -0.0590   -0.0480   -0.0271         0    0.0271    0.0480    0.0590
  -0.0324   -0.0263   -0.0149         0    0.0149    0.0263    0.0324
  -0.0164   -0.0133   -0.0075         0    0.0075    0.0133    0.0164
  -0.0077   -0.0062   -0.0035         0    0.0035    0.0062    0.0077
 Columns 15 through 21
   0.0077    0.0067    0.0052    0.0036    0.0023    0.0013    0.0007
   0.0165    0.0144    0.0111    0.0077    0.0048    0.0028    0.0014
   0.0326    0.0284    0.0220    0.0152    0.0096    0.0054    0.0028
   0.0594    0.0518    0.0400    0.0278    0.0174    0.0099    0.0052
   0.0999    0.0872    0.0674    0.0467    0.0293    0.0167    0.0087
   0.1552    0.1353    0.1046    0.0725    0.0455    0.0259    0.0135
   0.2224    0.1940    0.1499    0.1040    0.0652    0.0372    0.0193
   0.2943    0.2567    0.1984    0.1376    0.0863    0.0492    0.0256
   0.3595    0.3135    0.2423    0.1680    0.1054    0.0601    0.0312
   0.4053    0.3535    0.2732    0.1895    0.1188    0.0677    0.0352
   0.4218    0.3679    0.2843    0.1972    0.1237    0.0705    0.0366
```

```
    0.4053    0.3535    0.2732    0.1895    0.1188    0.0677    0.0352
    0.3595    0.3135    0.2423    0.1680    0.1054    0.0601    0.0312
    0.2943    0.2567    0.1984    0.1376    0.0863    0.0492    0.0256
    0.2224    0.1940    0.1499    0.1040    0.0652    0.0372    0.0193
    0.1552    0.1353    0.1046    0.0725    0.0455    0.0259    0.0135
    0.0999    0.0872    0.0674    0.0467    0.0293    0.0167    0.0087
    0.0594    0.0518    0.0400    0.0278    0.0174    0.0099    0.0052
    0.0326    0.0284    0.0220    0.0152    0.0096    0.0054    0.0028
    0.0165    0.0144    0.0111    0.0077    0.0048    0.0028    0.0014
    0.0077    0.0067    0.0052    0.0036    0.0023    0.0013    0.0007
```

最后利用函数[px,py] = gradient(z,.2,.2)求解数值梯度(x 和 y 方向上的间距均为 2)，可得：

```
px =
  Columns 1 through 7
   -0.0031   -0.0040   -0.0058   -0.0074   -0.0078   -0.0063   -0.0023
   -0.0066   -0.0085   -0.0124   -0.0157   -0.0167   -0.0135   -0.0050
   -0.0131   -0.0168   -0.0245   -0.0310   -0.0330   -0.0266   -0.0098
   -0.0238   -0.0307   -0.0446   -0.0566   -0.0601   -0.0484   -0.0179
   -0.0401   -0.0516   -0.0751   -0.0951   -0.1011   -0.0815   -0.0300
   -0.0623   -0.0801   -0.1165   -0.1477   -0.1570   -0.1265   -0.0467
   -0.0893   -0.1148   -0.1670   -0.2117   -0.2250   -0.1813   -0.0669
   -0.1181   -0.1518   -0.2210   -0.2802   -0.2977   -0.2399   -0.0885
   -0.1443   -0.1855   -0.2699   -0.3422   -0.3636   -0.2930   -0.1081
   -0.1627   -0.2091   -0.3043   -0.3858   -0.4100   -0.3303   -0.1218
   -0.1693   -0.2176   -0.3168   -0.4016   -0.4267   -0.3438   -0.1268
   -0.1627   -0.2091   -0.3043   -0.3858   -0.4100   -0.3303   -0.1218
   -0.1443   -0.1855   -0.2699   -0.3422   -0.3636   -0.2930   -0.1081
   -0.1181   -0.1518   -0.2210   -0.2802   -0.2977   -0.2399   -0.0885
   -0.0893   -0.1148   -0.1670   -0.2117   -0.2250   -0.1813   -0.0669
   -0.0623   -0.0801   -0.1165   -0.1477   -0.1570   -0.1265   -0.0467
   -0.0401   -0.0516   -0.0751   -0.0951   -0.1011   -0.0815   -0.0300
   -0.0238   -0.0307   -0.0446   -0.0566   -0.0601   -0.0484   -0.0179
   -0.0131   -0.0168   -0.0245   -0.0310   -0.0330   -0.0266   -0.0098
   -0.0066   -0.0085   -0.0124   -0.0157   -0.0167   -0.0135   -0.0050
   -0.0031   -0.0040   -0.0058   -0.0074   -0.0078   -0.0063   -0.0023
  Columns 8 through 14
    0.0037    0.0104    0.0156    0.0176    0.0156    0.0104    0.0037
    0.0079    0.0222    0.0334    0.0376    0.0334    0.0222    0.0079
    0.0156    0.0438    0.0659    0.0743    0.0659    0.0438    0.0156
    0.0285    0.0797    0.1200    0.1353    0.1200    0.0797    0.0285
    0.0480    0.1341    0.2019    0.2276    0.2019    0.1341    0.0480
    0.0745    0.2083    0.3135    0.3535    0.3135    0.2083    0.0745
    0.1067    0.2985    0.4493    0.5066    0.4493    0.2985    0.1067
    0.1412    0.3950    0.5945    0.6703    0.5945    0.3950    0.1412
    0.1725    0.4824    0.7261    0.8187    0.7261    0.4824    0.1725
    0.1945    0.5439    0.8187    0.9231    0.8187    0.5439    0.1945
    0.2024    0.5661    0.8521    0.9608    0.8521    0.5661    0.2024
    0.1945    0.5439    0.8187    0.9231    0.8187    0.5439    0.1945
    0.1725    0.4824    0.7261    0.8187    0.7261    0.4824    0.1725
    0.1412    0.3950    0.5945    0.6703    0.5945    0.3950    0.1412
    0.1067    0.2985    0.4493    0.5066    0.4493    0.2985    0.1067
    0.0745    0.2083    0.3135    0.3535    0.3135    0.2083    0.0745
```

```
    0.0480    0.1341    0.2019    0.2276    0.2019    0.1341    0.0480
    0.0285    0.0797    0.1200    0.1353    0.1200    0.0797    0.0285
    0.0156    0.0438    0.0659    0.0743    0.0659    0.0438    0.0156
    0.0079    0.0222    0.0334    0.0376    0.0334    0.0222    0.0079
    0.0037    0.0104    0.0156    0.0176    0.0156    0.0104    0.0037
  Columns 15 through 21
   -0.0023   -0.0063   -0.0078   -0.0074   -0.0058   -0.0040   -0.0031
   -0.0050   -0.0135   -0.0167   -0.0157   -0.0124   -0.0085   -0.0066
   -0.0098   -0.0266   -0.0330   -0.0310   -0.0245   -0.0168   -0.0131
   -0.0179   -0.0484   -0.0601   -0.0566   -0.0446   -0.0307   -0.0238
   -0.0300   -0.0815   -0.1011   -0.0951   -0.0751   -0.0516   -0.0401
   -0.0467   -0.1265   -0.1570   -0.1477   -0.1165   -0.0801   -0.0623
   -0.0669   -0.1813   -0.2250   -0.2117   -0.1670   -0.1148   -0.0893
   -0.0885   -0.2399   -0.2977   -0.2802   -0.2210   -0.1518   -0.1181
   -0.1081   -0.2930   -0.3636   -0.3422   -0.2699   -0.1855   -0.1443
   -0.1218   -0.3303   -0.4100   -0.3858   -0.3043   -0.2091   -0.1627
   -0.1268   -0.3438   -0.4267   -0.4016   -0.3168   -0.2176   -0.1693
   -0.1218   -0.3303   -0.4100   -0.3858   -0.3043   -0.2091   -0.1627
   -0.1081   -0.2930   -0.3636   -0.3422   -0.2699   -0.1855   -0.1443
   -0.0885   -0.2399   -0.2977   -0.2802   -0.2210   -0.1518   -0.1181
   -0.0669   -0.1813   -0.2250   -0.2117   -0.1670   -0.1148   -0.0893
   -0.0467   -0.1265   -0.1570   -0.1477   -0.1165   -0.0801   -0.0623
   -0.0300   -0.0815   -0.1011   -0.0951   -0.0751   -0.0516   -0.0401
   -0.0179   -0.0484   -0.0601   -0.0566   -0.0446   -0.0307   -0.0238
   -0.0098   -0.0266   -0.0330   -0.0310   -0.0245   -0.0168   -0.0131
   -0.0050   -0.0135   -0.0167   -0.0157   -0.0124   -0.0085   -0.0066
   -0.0023   -0.0063   -0.0078   -0.0074   -0.0058   -0.0040   -0.0031
py =
  Columns 1 through 7
   -0.0038   -0.0073   -0.0129   -0.0206   -0.0296   -0.0383   -0.0440
   -0.0054   -0.0104   -0.0182   -0.0291   -0.0419   -0.0543   -0.0622
   -0.0093   -0.0179   -0.0314   -0.0501   -0.0723   -0.0935   -0.1072
   -0.0146   -0.0281   -0.0494   -0.0787   -0.1135   -0.1468   -0.1683
   -0.0208   -0.0400   -0.0702   -0.1119   -0.1614   -0.2088   -0.2394
   -0.0266   -0.0512   -0.0898   -0.1432   -0.2064   -0.2670   -0.3062
   -0.0302   -0.0581   -0.1020   -0.1626   -0.2344   -0.3033   -0.3478
   -0.0297   -0.0573   -0.1005   -0.1602   -0.2309   -0.2988   -0.3426
   -0.0241   -0.0464   -0.0814   -0.1297   -0.1870   -0.2420   -0.2775
   -0.0135   -0.0261   -0.0457   -0.0729   -0.1051   -0.1360   -0.1559
         0         0         0         0         0         0         0
    0.0135    0.0261    0.0457    0.0729    0.1051    0.1360    0.1559
    0.0241    0.0464    0.0814    0.1297    0.1870    0.2420    0.2775
    0.0297    0.0573    0.1005    0.1602    0.2309    0.2988    0.3426
    0.0302    0.0581    0.1020    0.1626    0.2344    0.3033    0.3478
    0.0266    0.0512    0.0898    0.1432    0.2064    0.2670    0.3062
    0.0208    0.0400    0.0702    0.1119    0.1614    0.2088    0.2394
    0.0146    0.0281    0.0494    0.0787    0.1135    0.1468    0.1683
    0.0093    0.0179    0.0314    0.0501    0.0723    0.0935    0.1072
    0.0054    0.0104    0.0182    0.0291    0.0419    0.0543    0.0622
    0.0038    0.0073    0.0129    0.0206    0.0296    0.0383    0.0440
  Columns 8 through 14
```

```
  -0.0436   -0.0355   -0.0200         0    0.0200    0.0355    0.0436
  -0.0617   -0.0503   -0.0283         0    0.0283    0.0503    0.0617
  -0.1064   -0.0867   -0.0489         0    0.0489    0.0867    0.1064
  -0.1670   -0.1360   -0.0767         0    0.0767    0.1360    0.1670
  -0.2376   -0.1935   -0.1091         0    0.1091    0.1935    0.2376
  -0.3039   -0.2474   -0.1395         0    0.1395    0.2474    0.3039
  -0.3451   -0.2810   -0.1584         0    0.1584    0.2810    0.3451
  -0.3400   -0.2768   -0.1561         0    0.1561    0.2768    0.3400
  -0.2754   -0.2242   -0.1264         0    0.1264    0.2242    0.2754
  -0.1547   -0.1260   -0.0710         0    0.0710    0.1260    0.1547
        0         0         0         0         0         0         0
   0.1547    0.1260    0.0710         0   -0.0710   -0.1260   -0.1547
   0.2754    0.2242    0.1264         0   -0.1264   -0.2242   -0.2754
   0.3400    0.2768    0.1561         0   -0.1561   -0.2768   -0.3400
   0.3451    0.2810    0.1584         0   -0.1584   -0.2810   -0.3451
   0.3039    0.2474    0.1395         0   -0.1395   -0.2474   -0.3039
   0.2376    0.1935    0.1091         0   -0.1091   -0.1935   -0.2376
   0.1670    0.1360    0.0767         0   -0.0767   -0.1360   -0.1670
   0.1064    0.0867    0.0489         0   -0.0489   -0.0867   -0.1064
   0.0617    0.0503    0.0283         0   -0.0283   -0.0503   -0.0617
   0.0436    0.0355    0.0200         0   -0.0200   -0.0355   -0.0436
 Columns 15 through 21
   0.0440    0.0383    0.0296    0.0206    0.0129    0.0073    0.0038
   0.0622    0.0543    0.0419    0.0291    0.0182    0.0104    0.0054
   0.1072    0.0935    0.0723    0.0501    0.0314    0.0179    0.0093
   0.1683    0.1468    0.1135    0.0787    0.0494    0.0281    0.0146
   0.2394    0.2088    0.1614    0.1119    0.0702    0.0400    0.0208
   0.3062    0.2670    0.2064    0.1432    0.0898    0.0512    0.0266
   0.3478    0.3033    0.2344    0.1626    0.1020    0.0581    0.0302
   0.3426    0.2988    0.2309    0.1602    0.1005    0.0573    0.0297
   0.2775    0.2420    0.1870    0.1297    0.0814    0.0464    0.0241
   0.1559    0.1360    0.1051    0.0729    0.0457    0.0261    0.0135
        0         0         0         0         0         0         0
  -0.1559   -0.1360   -0.1051   -0.0729   -0.0457   -0.0261   -0.0135
  -0.2775   -0.2420   -0.1870   -0.1297   -0.0814   -0.0464   -0.0241
  -0.3426   -0.2988   -0.2309   -0.1602   -0.1005   -0.0573   -0.0297
  -0.3478   -0.3033   -0.2344   -0.1626   -0.1020   -0.0581   -0.0302
  -0.3062   -0.2670   -0.2064   -0.1432   -0.0898   -0.0512   -0.0266
  -0.2394   -0.2088   -0.1614   -0.1119   -0.0702   -0.0400   -0.0208
  -0.1683   -0.1468   -0.1135   -0.0787   -0.0494   -0.0281   -0.0146
  -0.1072   -0.0935   -0.0723   -0.0501   -0.0314   -0.0179   -0.0093
  -0.0622   -0.0543   -0.0419   -0.0291   -0.0182   -0.0104   -0.0054
  -0.0440   -0.0383   -0.0296   -0.0206   -0.0129   -0.0073   -0.0038
```

为了图形显示所给函数的数值梯度，调用函数 contour(v,v,z)、hold on、quiver(px,py)和 hold off，如图 7-6 所示。

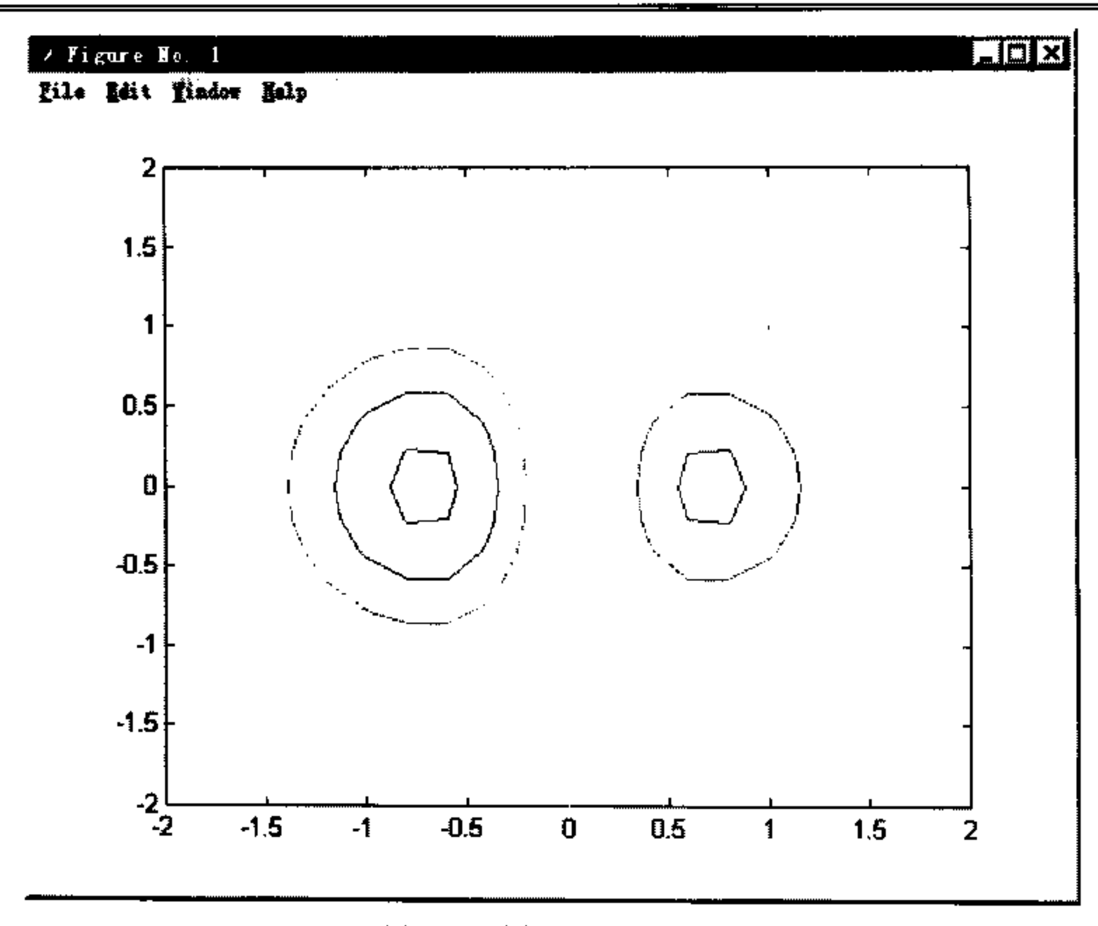


图 7-6　所给函数的数值梯度

7.3　相关

1．相关系数

名称：corrcoef

相关系数。

语法：该函数有如下两种表达形式：

- S = corrcoef (X)
- S = corrcoef (x, y)

描述：S = corrcoef (X)返回 X 的相关系数矩阵 S，其中 X 矩阵的行向量为观察对象，而列向量为变量。

矩阵 S = corrcoef (X)通过下式与协方差矩阵 C = cov(X)发生联系：

$$S(i,j)=\frac{C(i,j)}{\sqrt{C(i,i)C(j,j)}}\text{。}$$

S = corrcoef (x, y)返回与 corrcoef ([x, y])相同的结果，其中 x 和 y 均为列向量。

举例：

例如对于任意一个 3×3 矩阵：

X =

```
1    2    3
4    5    6
2    7    5
```

在 MATLAB 命令窗口中输入 S = corrcoef (X)，可得 X 的相关系数矩阵：

```
S =
    1.0000    0.4336    0.9286
    0.4336    1.0000    0.7370
    0.9286    0.7370    1.0000
```

另外，对于任意的列向量 x 和 y：

```
x =
     3
     7
     1
y =
    11
     4
     0
```

利用函数 S = corrcoef (x, y)可以求得 x 和 y 之间的相关系数，结果为：

```
S =
    1.0000    0.1764
    0.1764    1.0000
```

2. 协方差矩阵

名称：cov

协方差矩阵。

语法：该函数有如下两种表达形式：

- C = cov (X)
- C = cov (x, y)

描述：当 X 是一个向量时，C = cov (X)返回该向量中元素的方差。

当 X 是一个矩阵(其行向量为观察对象，而列向量为变量)时，C = cov (X)返回协方差矩阵。利用函数 diag(cov(x))可以得到由每一列元素的方差所组成的向量，而 sqrt(diag(cov(x)))返回一个标准偏差向量。

C = cov (x, y)等价于 cov([x y])，其中 x 和 y 是等长度的列向量。

举例：

考虑如下所示的矩阵：

```
A =
    -1    1    2
    -2    3    1
     4    0    3
```

首先利用函数 v = diag(cov(A))'得到 A 中每一列元素的方差所组成的向量：

v = 10.3333　　2.3333　　1.0000。

利用函数 C = cov (X)可以得到协方差矩阵：

```
C =
   10.3333   -4.1667    3.0000
   -4.1667    2.3333   -1.5000
    3.0000   -1.5000    1.0000。
```

7.4　滤波和卷积

1.　卷积和多项式相乘

名称：conv

卷积和多项式相乘。

语法：w = conv(u,v)

描述：卷积和解卷积是信号处理中常用的数学工具。向量的卷积和解卷积对应于多项式的乘法和除法。函数 conv 和 deconv 分别为卷积函数和解卷积函数。此处先介绍卷积函数conv。

w = conv(u,v)返回向量 u 和向量 v 的卷积。从代数上来讲，所谓 u 和 v 的卷积，实际上就是分别以 u 和 v 中的元素为系数的两个多项式的乘积。

令 m=length(u)、n = length(v)，则 w 就是长度为 m+n-1 的向量，其第 k 个元素为：

$$w(k) = \sum_j u(i)v(k+1-j)\text{。}$$

上式右端项中的求和运算是针对所有使 u(j)和 v(k+1-j)合法的 j 的取值而言的。当 m=n 时，从上式可以得到：

w(1) = u(1)*v(1)
w(2) = u(1)*v(2)+u(2)*v(1)
w(3) = u(1)*v(3)+u(2)*v(2)+u(3)*v(1)
...
w(n) = u(1)*v(n)+u(2)*v(n-1)+ ... +u(n)*v(1)
...
w(2*n-1) = u(n)*v(n)

举例：

本例求解多项式 $a(s)=s^2+2s+3$ 和 $b(s)=4s^2+5s+6$ 的乘积 c(s)。其中：

a = 1　2　3，b = 4　5　6。

利用函数 c = conv(a,b)可以得到：c = 4　13　28　27　18。

因此多项式 a(s)和 b(s)的乘积为 $c(s)=4s^4+13s^3+28s^2+27s+18$。

2.　二维卷积

名称：conv2

二维卷积。

语法：该函数有如下几种表达形式：

- C = conv2(A,B)
- C = conv2(hcol,hrow,A)
- C = conv2(...,'shape')

描述：C = conv2 (A, B)返回矩阵 A 和 B 的二维卷积。如果 A 和 B 中有一个是二维滤波器矩阵，则另一个必将被滤波为二维。如果矩阵 A 的维数为 ma×na，矩阵 B 的维数为 mb×nb，则矩阵 C 的维数为[ma+mb-1,na+nb-1]。

函数 C = conv2 (hcol, hrow, A)将矩阵 A 在列的方向上与向量 hcol 进行卷积，而在行的方向上与向量 hrow 进行卷积。

函数 C = conv2 (..., 'shape')依参量 shape 的不同取值，返回不同的二维卷积的子分量。

当 shape 取为 full (缺省)时，函数 conv2 (..., 'shape')返回全部二维卷积。

当 shape 取为 same 时，函数 conv2 (..., 'shape')返回与矩阵 A 尺寸相同的二维卷积的中间部分。

当 shape 取 valid 时，函数 conv2 (..., 'shape')返回卷积中不是由零边界计算得到的部分。

举例:

在图像处理中，寻找 Sobel 边界的操作实质上是一个输入数组与指定矩阵的二维卷积。该指定矩阵为：

```
s =
     1     2     1
     0     0     0
    -1    -2    -1
```

首先利用函数 A = zeros(10)初始化数组 A，可得：

```
A =
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

然后利用函数 A(3:7,3:7) = ones(5)为数组 A 赋值，结果为：

```
A =
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     1     1     1     1     1     0     0     0
     0     0     1     1     1     1     1     0     0     0
     0     0     1     1     1     1     1     0     0     0
     0     0     1     1     1     1     1     0     0     0
     0     0     1     1     1     1     1     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

最后利用函数 H = conv2(A,s)寻找 Sobel 边界(即求解二维卷积)，结果为：

```
H =
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     1     3     4     4     4     3     1     0     0     0
     0     0     1     3     4     4     4     3     1     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0    -1    -3    -4    -4    -4    -3    -1     0     0     0
     0     0    -1    -3    -4    -4    -4    -3    -1     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
```

为了直观起见，利用函数 mesh(H)将结果在窗口中显示，如图 7-7 所示。

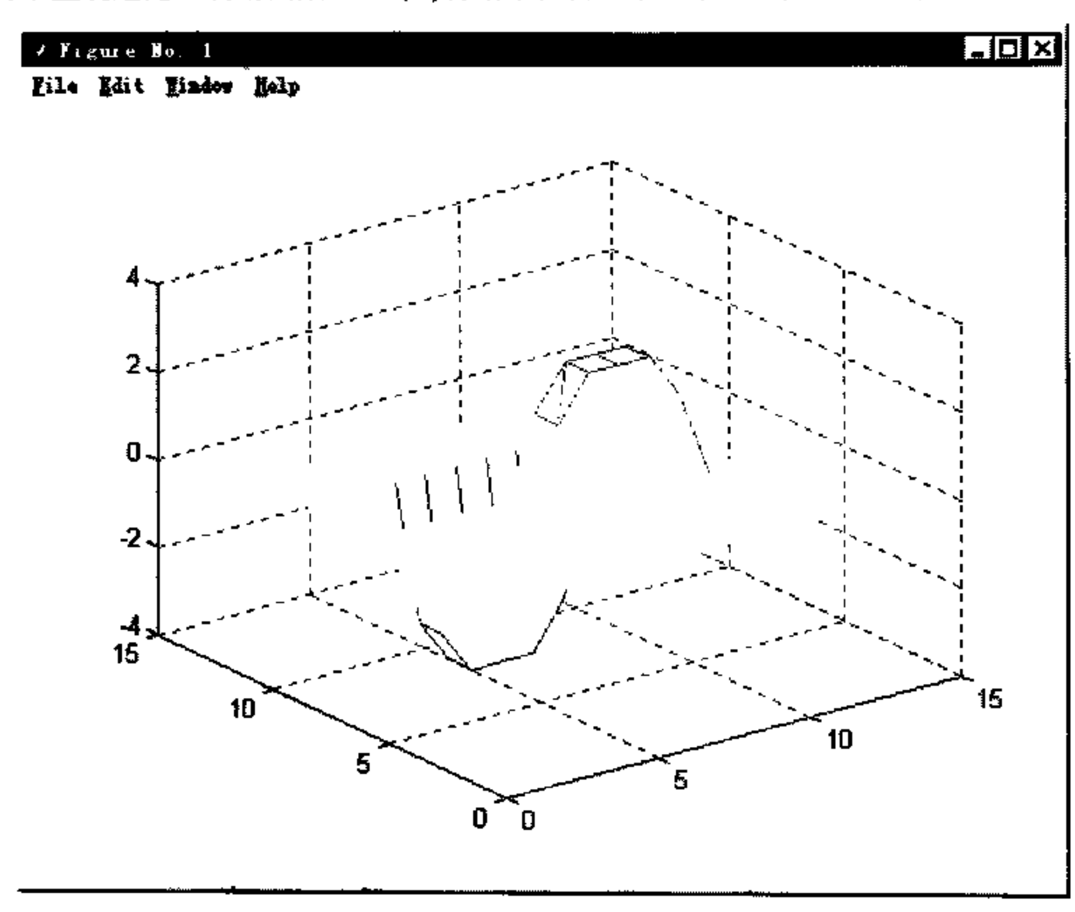


图 7-7 Sobel 边界图

下面的操作则是首先显示 A 的垂直边界，然后再同时显示垂直和水平边界：
在 MATLAB 命令窗口中输入 V = conv2(A,s')得到：

V =

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	-1	-1	0	0	0
0	0	3	3	0	0	0	-3	-3	0	0	0
0	0	4	4	0	0	0	-4	-4	0	0	0
0	0	4	4	0	0	0	-4	-4	0	0	0
0	0	4	4	0	0	0	-4	-4	0	0	0
0	0	3	3	0	0	0	-3	-3	0	0	0
0	0	1	1	0	0	0	-1	-1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

然后使用函数 mesh(V)绘制 A 的垂直边界，如图 7-8 所示。

最后使用函数 mesh(sqrt(H.^2+V.^2)) 同时绘制垂直和水平边界，如图 7-9 所示。

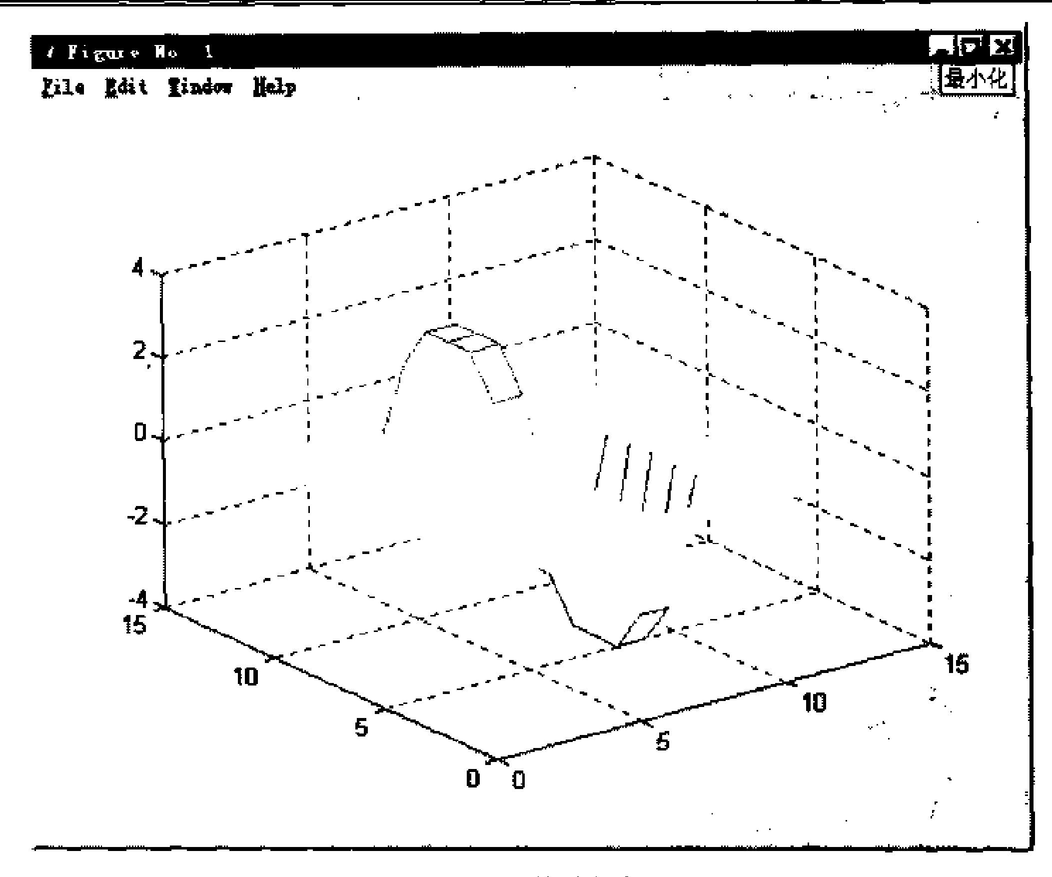


图 7-8 A 的垂直边界

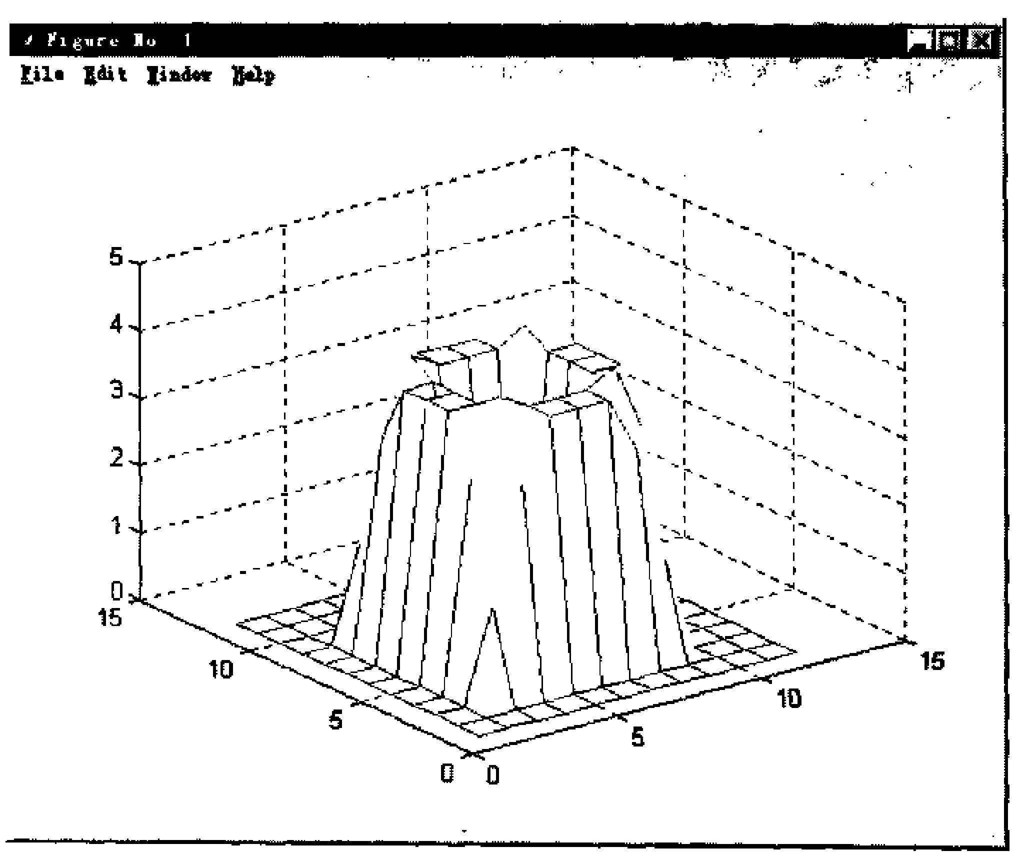


图 7-9 同时显示 A 的垂直边界和水平边界

3. 解卷积和多项式相除

名称：deconv

解卷积和多项式相除。

语法：[q, r] = deconv (v, u)

描述：如前所述，解卷积是信号处理中常用的数学工具。向量解卷积对应于多项式的除法。此处介绍解卷积函数 deconv。

函数[q, r] = deconv (v, u)使用长除法从向量 v 中解卷积向量 u。求得的商返回到向量 q 中，余数则返回到向量 r 中，最后满足：v = conv(u,q)+r。

举例：

例如对于向量 u 和 v：

u = 1 2 3 4 和 v =10 20 30。

利用函数 c = conv(u,v)可以求得二者的卷积为：

c = 10 40 100 160 170 120。

此时利用解卷积函数[q,r] = deconv(c,u)可以恢复向量 v，结果为：

q = 10 20 30

r = 0 0 0 0 0 0。

4. 利用 IIR 或 IIR 滤波器对数据进行滤波

名称：filter

利用 IIR 或 IIR 滤波器对数据进行滤波。

语法：该函数有如下几种表达形式：

- y = filter (b, a, X)
- [y, zf] = filter (b, a, X)
- [y, zf] = filter (b, a, X, zi)
- y = filter (b, a, X, zi, dim)
- [...] = filter (b, a, X, [], dim)

描述：滤波器函数既可以对实数进行滤波，也可以对复数进行滤波。

函数 y = filter (b, a, X)使用以向量 b 作为分子、以向量 a 作为分母的滤波器对向量 X 中的数据进行滤波。如果 a(1)不等于 1，则滤波器使用 a(1)来规格化滤波器系数。如果 a(1)等于 0，则滤波器返回一个出错信息。

如果 X 是一个矩阵，则滤波器对 X 的列向量进行滤波作用。

[y, zf] = filter (b, a, X)返回滤波器迟滞的最后条件：向量 zf。

[y, zf] = filter (b, a, X, zi)接收滤波器迟滞的初始条件：向量 zi，并返回滤波器迟滞的最后条件：向量 zf。

y = filter (b, a, X, zi, dim)与[...] = filter (b, a, X, [], dim)则是越过维数 dim 进行滤波操作。

举例：

例如对于任意的一个矩阵：

```
X =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
```

```
    17      18      19      20
```

令向量 a 和 b 分别为：

a =5　3　8 和 b = 4　7　2。

利用函数 y = filter (b, a, X)对 X 中的数据进行滤波，可得：

```
y =
     0.8000     1.6000     2.4000     3.2000
     4.9200     6.6400     8.3600    10.0800
    10.3680    10.6560    10.9440    11.2320
    10.9072    10.5824    10.2576     9.9328
    12.2669    14.6010    16.9350    19.2691
```

利用函数[y, zf] = filter (b, a, X)同时还可以得到滤波器迟滞的最后条件(向量 zf)，如下所示：

```
zf =
     4.1884     5.1076     6.0268     6.9460
   -12.8270   -16.1615   -19.4961   -22.8306
```

5. 二维数字滤波

名称：filter2

二维数字滤波。

语法：该函数有如下两种表达形式：

- Y = filter2 (h, X)
- Y = filter2 (h, X, shape)

描述：函数 Y = filter2 (h, X)使用矩阵 h 中的二维 FIR 滤波器对 X 中的数据进行滤波。

Y = filter2 (h, X, shape)依 shape 取值的不同而返回不同的滤波结果。

若 shape 取 full，则返回全部二维相关矩阵。在这种情况下，矩阵 Y 要大于 X。

若 shape 取 same (缺省情况)，则返回相关矩阵的中间部分。在这种情况下，矩阵 Y 和矩阵 X 的尺寸相同。

若 shape 取 valid，则返回相关矩阵中不是由零边界计算得到的部分。在这种情况下，矩阵 Y 要小于 X。

举例：

例如对于任意的一个矩阵：

```
X =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
    17    18    19    20
```

取矩阵 h 为：

```
h =
     1     2     3
     1     1     1
     2     5     6
```

利用函数 Y = filter2 (h, X)进行二维数字滤波，可得：

```
Y =
    64    88   104    61
   124   166   188   108
   196   254   276   156
```

```
  268   342   364   204
  103   140   149    86
```

7.5 傅里叶变换

1. 模

名称：abs

求绝对值和复数的模。

语法：Y = abs (X)

描述：Y = abs (X)返回 X 中每个元素的绝对值。

如果 X 是复数，则返回复数的模：abs(X) = sqrt(real(X).^2 + imag(X).^2) 。

举例：

abs(-5) = 5
abs(3+4i) = 5

2. 相角

名称：angle

相角。

语法：P = angle (Z)

描述：P = angle (Z)返回复数数组 Z 中每个元素的相角，单位为弧度。

举例：

例如对于复数矩阵：

```
Z =
   1.0000 - 1.0000i   2.0000 + 1.0000i   3.0000 - 1.0000i   4.0000 + 1.0000i
   1.0000 + 2.0000i   2.0000 - 2.0000i   3.0000 + 2.0000i   4.0000 - 2.0000i
   1.0000 - 3.0000i   2.0000 + 3.0000i   3.0000 - 3.0000i   4.0000 + 3.0000i
   1.0000 + 4.0000i   2.0000 - 4.0000i   3.0000 + 4.0000i   4.0000 - 4.0000i
```

执行函数 P = angle (Z)可以得到 Z 中每个元素的相角，结果为：

```
P =
   -0.7854    0.4636   -0.3218    0.2450
    1.1071   -0.7854    0.5880   -0.4636
   -1.2490    0.9828   -0.7854    0.6435
    1.3258   -1.1071    0.9273   -0.7854
```

3. 按复共轭把复数进行分类

名称：cplxpair

按复共轭把复数进行分类。

语法：该函数有如下几种表达形式：

- B = cplxpair (A)
- B = cplxpair (A, tol)
- B = cplxpair (A, [], dim)
- B = cplxpair (A, tol, dim)

描述：B = cplxpair (A)按复共轭把复数数组中的元素进行分类和排列。

如果 A 是一个向量，则 cplxpair (A)返回按复共轭对 A 重新进行排列后得到的向量。

如果 A 是一个矩阵，则 cplxpair (A)返回一个按复共轭对 A 中的列向量重新进行排列后得到的矩阵。

B = cplxpair (A, tol)重载缺省的允许值，并进行重排。

B = cplxpair (A, [], dim)沿着由标量 dim 指定的维数对 A 进行重排。

B = cplxpair (A, tol, dim)沿着由标量 dim 指定的维数对数组 A 进行重排，并重载缺省的允许值。

举例：

例如对于复数向量：

A = 4.0000 + 2.0000i　4.0000 - 2.0000i　3.0000 + 6.0000i　3.0000 - 6.0000i，

利用函数 B = cplxpair (A)可以按复共轭把该数组中的元素进行分类，结果为：

B = 3.0000 - 6.0000i　3.0000 + 6.0000i　4.0000 - 2.0000i　4.0000 + 2.0000i。

例如对于复数矩阵：

```
A =
    1.0000 + 3.0000i    2.0000 - 4.0000i
    1.0000 - 3.0000i    2.0000 + 4.0000i
```

利用函数 B = cplxpair (A)可以按复共轭把该数组中的元素进行分类，结果为：

```
B =
    1.0000 - 3.0000i    2.0000 - 4.0000i
    1.0000 + 3.0000i    2.0000 + 4.0000i
```

利用函数 B = cplxpair (A, 0.9)将缺省的允许值重载为 0.9，并对 A 进行重排，结果为：

```
B =
    1.0000 - 3.0000i    2.0000
    1.0000 + 3.0000i    2.0000
```

4. 一维快速傅里叶变换

名称：fft

一维快速傅里叶变换。

语法：该函数有如下几种表达形式：

- Y = fft (X)
- Y = fft (X, n)
- Y = fft (X, [], dim)
- Y = fft (X, n, dim)

描述：傅里叶分析把信号分解成不同频率的正弦函数的叠加。傅里叶变换是信号处理的最重要、最基本的工具之一。对于离散信号采用离散傅里叶变换(DFT)进行分析。

快速傅里叶变换(FFT)是离散傅里叶变换的一种快速算法。正是由于有了快速傅里叶变换，傅里叶分析才被广泛应用在滤波、卷积、频域分析和功率谱估计上。

Y = fft (X)返回采用快速傅里叶变换(FFT)计算得到的向量 X 的离散傅里叶变换。

如果 X 是一个矩阵，fft (X)则返回 X 中每一列的傅里叶变换。

函数 Y = fft (X, n)返回 n 点快速傅里叶变换。其中参数 n 用来指定变换的点数。当数据 X 中的点数少于 n 点时，MATLAB 会自动补零使得总点数为 n；当数据 X 中的点数多于 n 点时，MATLAB 会自动截断输入数据；当不输入参数时，MATLAB 使用 X 的点数进行计算。而且当 fft 的计算点数为 2 的整数次幂时，计算速度最快。

Y = fft (X, [], dim)和 Y = fft (X, n, dim)则越过维数 dim 执行傅里叶变换。

举例：

傅里叶变换常用来寻找噪音时域信号中所包含的某一信号的频率分量。

例如考虑在 100Hz 频率条件下所采集的信号数据，同时形成了包含 50Hz 和 120Hz 的信号，并夹杂进一些随机噪音。

首先利用下列函数产生包含随机噪音的信号数据：

t = 0:0.001:0.6，x = sin(2*pi*50*t)+sin(2*pi*120*t)和 y = x + 2*randn(size(t))。

并利用函数 plot(y(1:50))将原始信号在窗口中显示出来，如图 7-10 所示。

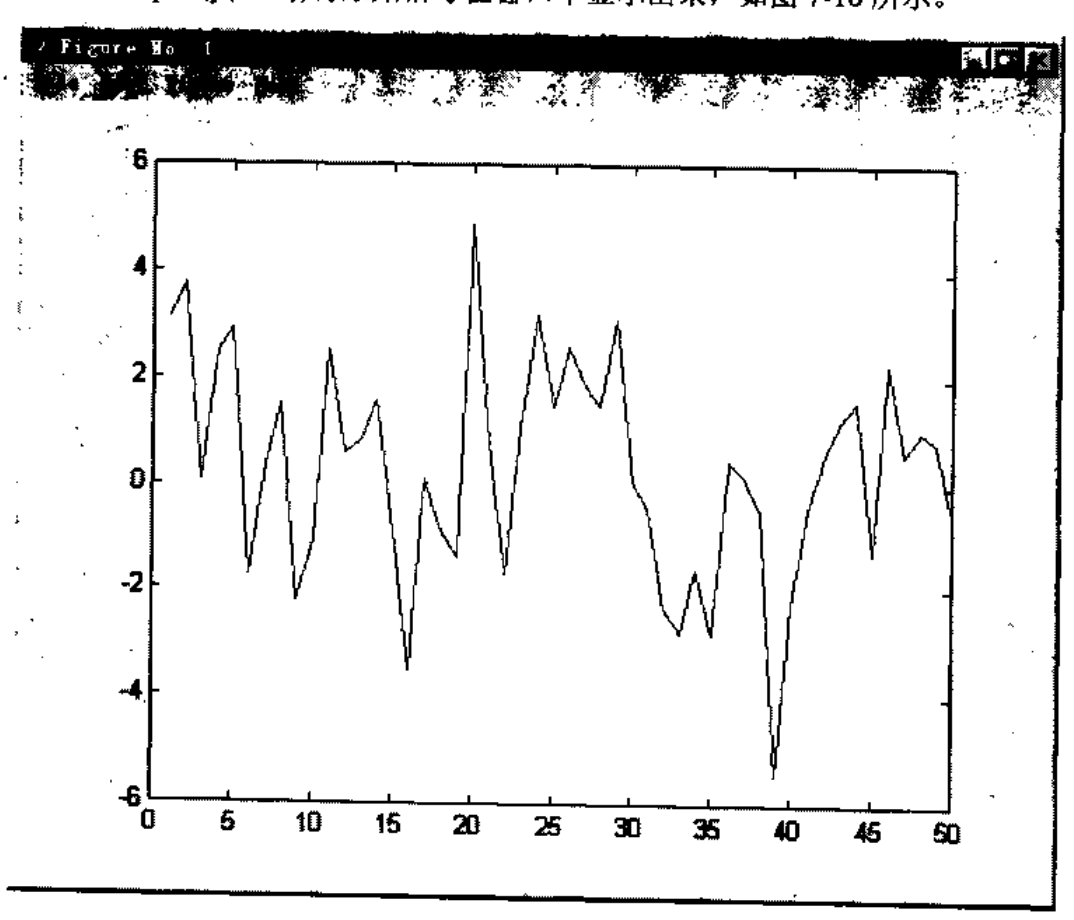


图 7-10 原始数据图

直接从原始数据图中来识别某一频率分量将会非常困难。现在利用函数 Y = fft(y,512)，即 512 点快速傅里叶变换，就可以通过将信号转换到频域，从而找出噪音信号的离散傅里叶变换。

利用函数 Pyy = Y.* conj(Y) / 512 和 f = 1000*(0:256)/512，可以估计不同频率的能量。

最后利用函数 plot(f,Pyy(1:257))可以绘制出频域内的曲线，如图 7-11 所示。

5. 二维快速傅里叶变换

名称：fft2

二维快速傅里叶变换。

语法：该函数有如下两种表达形式：

- Y = fft2 (X)
- Y = fft2 (X, m, n)

描述：Y = fft2 (X)返回二维快速傅里叶变换。而且 Y 和 X 有相同的尺寸。

函数 Y = fft2 (X, m, n)在进行二维快速傅里叶变换之前，将 X 通过截断或补零变为一个 m×n 的数组。

举例：

可参见函数 fft 中的例子。

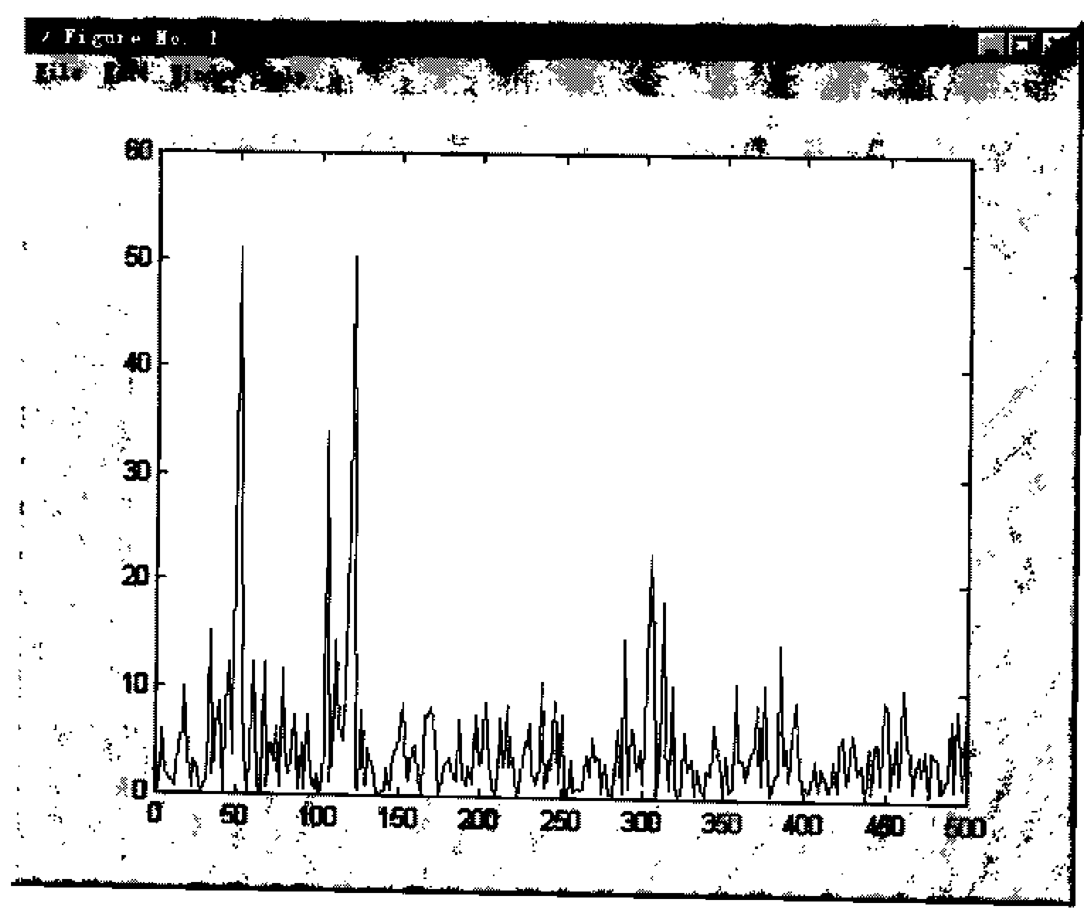


图 7-11　频域曲线

6. *移谱*

名称：fftshift

将快速傅里叶变换的 DC 分量移到谱中心。

语法：Y = fftshift (X)

描述：函数 Y = fftshift (X)通过将零频分量移动到数组中间，重新对函数 fft、fft2 和 fftn 的输出结果进行重排。

举例：

例如对于任意矩阵 X，

```
X =
     1     2     3
     1     4     5
     4     5     6
     6     8     9
     1     2     5
     0     0     3
     5     6     7
```

经过 Y = fft2(X)变换后的矩阵 Y 中的第一个元素满足：Y(1,1)=sum(sum(X))，而且信号

的 DC 分量位于二维快速傅里叶变换的左上角，如下所示：

```
Y =
  83.0000                -14.5000 + 9.5263i -14.5000 - 9.5263i
  -8.4777 -11.9527i   -1.8852 - 0.3823i    2.9951 + 3.5096i
   2.8802 +24.7335i    1.3037 + 1.4692i   -2.9076 + 2.4305i
 -14.9025 - 1.7709i    1.9125 + 1.5623i    2.5814 + 0.1733i
 -14.9025 + 1.7709i    2.5814 - 0.1733i    1.9125 - 1.5623i
   2.8802 -24.7335i   -2.9076 - 2.4305i    1.3037 - 1.4692i
  -8.4777 +11.9527i    2.9951 - 3.5096i   -1.8852 + 0.3823i
```

此时执行函数 Z = fftshift(Y)，就可以将 DC 分量移动到矩阵的中间，结果为：

```
Z =
   1.9125 - 1.5623i -14.9025 + 1.7709i    2.5814 - 0.1733i
   1.3037 - 1.4692i    2.8802 -24.7335i   -2.9076 - 2.4305i
  -1.8852 + 0.3823i   -8.4777 +11.9527i    2.9951 - 3.5096i
 -14.5000 - 9.5263i   83.0000             -14.5000 + 9.5263i
   2.9951 + 3.5096i   -8.4777 -11.9527i   -1.8852 - 0.3823i
  -2.9076 + 2.4305i    2.8802 +24.7335i    1.3037 + 1.4692i
   2.5814 + 0.1733i -14.9025 - 1.7709i    1.9125 + 1.5623i
```

7.　一维逆快速傅里叶变换

名称：ifft

一维逆快速傅里叶变换。

语法：该函数有如下几种表达形式：

- y = ifft (X)
- y = ifft (X, n)
- y = ifft (X, [], dim)
- y = ifft (X, n, dim)

描述：y = ifft (X)返回向量 X 的逆快速傅里叶变换。

如果 X 是一个矩阵，ifft (X)返回 X 中每一列的逆傅里叶变换。

函数 y = ifft (X, n)返回 n 点逆快速傅里叶变换。

y = ifft (X, [], dim)和 y = ifft (X, n, dim) 则越过维数 dim 执行逆傅里叶变换。

举例：

利用 t = 0:0.02:0.6 和 x = sin(2*pi*50*t)+sin(2*pi*120*t)得到向量 x：

```
x =
  Columns 1 through 7
         0    0.5878   -0.9511    0.9511   -0.5878   -0.0000    0.5878
  Columns 8 through 14
   -0.9511    0.9511   -0.5878   -0.0000    0.5878   -0.9511    0.9511
  Columns 15 through 21
   -0.5878   -0.0000    0.5878   -0.9511    0.9511   -0.5878   -0.0000
  Columns 22 through 28
    0.5878   -0.9511    0.9511   -0.5878   -0.0000    0.5878   -0.9511
  Columns 29 through 31
    0.9511   -0.5878   -0.0000
```

首先利用函数 y = fft(x)进行快速傅里叶变换，得到：

```
y =
  Columns 1 through 4
  -0.0000              0.0034 - 0.0331i   0.0138 - 0.0671i   0.0323 - 0.1030i
  Columns 5 through 8
   0.0610 - 0.1422i   0.1035 - 0.1864i   0.1659 - 0.2383i   0.2597 - 0.3025i
  Columns 9 through 12
   0.4072 - 0.3870i   0.6583 - 0.5095i   1.1484 - 0.7158i   2.4061 - 1.1802i
  Columns 13 through 16
  10.2845 - 3.8090i  -8.4320 + 2.1832i  -3.9574 + 0.6063i  -3.1545 + 0.1600i
  Columns 17 through 20
  -3.1545 - 0.1600i  -3.9574 - 0.6063i  -8.4320 - 2.1832i  10.2845 + 3.8090i
  Columns 21 through 24
   2.4061 + 1.1802i   1.1484 + 0.7158i   0.6583 + 0.5095i   0.4072 + 0.3870i
  Columns 25 through 28
   0.2597 + 0.3025i   0.1659 + 0.2383i   0.1035 + 0.1864i   0.0610 + 0.1422i
  Columns 29 through 31
   0.0323 + 0.1030i   0.0138 + 0.0671i   0.0034 + 0.0331i
```

现在利用函数 x1 = ifft (y)就可以得到与 x 相同的向量 x1(只是在表达式中多了虚数项)：

```
x1 =
  Columns 1 through 4
  -0.0000 - 0.0000i   0.5878 + 0.0000i  -0.9511 - 0.0000i   0.9511 + 0.0000i
  Columns 5 through 8
  -0.5878 - 0.0000i  -0.0000 - 0.0000i   0.5878 + 0.0000i  -0.9511 - 0.0000i
  Columns 9 through 12
   0.9511 + 0.0000i  -0.5878 + 0.0000i  -0.0000 - 0.0000i   0.5878 + 0.0000i
  Columns 13 through 16
  -0.9511 - 0.0000i   0.9511 - 0.0000i  -0.5878 + 0.0000i  -0.0000 - 0.0000i
  Columns 17 through 20
   0.5878 + 0.0000i  -0.9511 - 0.0000i   0.9511 + 0.0000i  -0.5878 + 0.0000i
  Columns 21 through 24
  -0.0000 - 0.0000i   0.5878 + 0.0000i  -0.9511 - 0.0000i   0.9511 + 0.0000i
  Columns 25 through 28
  -0.5878 + 0.0000i  -0.0000 - 0.0000i   0.5878 + 0.0000i  -0.9511 - 0.0000i
  Columns 29 through 31
   0.9511 - 0.0000i  -0.5878 + 0.0000i  -0.0000 - 0.0000i
```

8. 二维逆快速傅里叶变换

名称：ifft2

二维逆快速傅里叶变换。

语法：该函数有如下两种表达形式：

- Y = ifft2 (X)
- Y = ifft2 (X, m, n)

描述：Y = ifft2 (X)返回矩阵 X 的二维逆快速傅里叶变换。

Y = ifft2 (X, m, n)返回矩阵 X 的 m×n 阶逆傅里叶变换。

举例：

可参见函数 ifft 中的例子。

9. 多维逆快速傅里叶变换

名称：ifftn

多维逆快速傅里叶变换。

语法：该函数有如下两种表达形式：

- Y = ifftn (X)
- Y = ifftn (X, siz)

描述：Y = ifftn (X)返回 N 维逆快速傅里叶变换。

Y = ifftn (X, siz) 在进行逆傅里叶变换之前，将 X 通过截断或补零变为一个尺寸为 siz 的数组。

举例：

可参见函数 ifft 中的例子。

10. 逆 FFT 平移

名称：ifftshift

逆 FFT 平移。

语法：ifftshift(X)

描述：当 X 是矩阵时，函数 ifftshift(X)将 X 的第一象限与第三象限进行交换、第二象限与第四象限进行交换。

举例：

例如对于任意矩阵 X，

```
X =
     1     2     3
     1     4     5
     4     5     6
     6     8     9
     1     2     5
     0     0     3
     5     6     7
```

执行函数 Y = fftshift(X)，就可以将 DC 分量移动到矩阵的中间，结果为：

```
Y =

     5     1     2
     3     0     0
     7     5     6
     3     1     2
     5     1     4
     6     4     5
     9     6     8
```

此时若执行函数 ifftshift(X)，可以得到：

```
ans =

     8     9     6
     2     5     1
     0     3     0
     6     7     5
```

```
    2     3     1
    4     5     1
    5     6     4
```

11. 最相邻的 2 的幂

名称：nextpow2

最相邻的 2 的幂。

语法：p = nextpow2 (A)

描述：p = nextpow2 (A)返回大于或等于 A 的绝对值的 2 的最小次幂。

如果 A 不是标量，则 nextpow2 (A)返回大于或等于 length(A)的 2 的最小次幂。

举例：

对于任何位于 513 和 1024 之间的整数 n，nextpow2(n)的返回值为 10。

对于一个 1×30 的向量 A，length(A)的值为 30，而 nextpow2(A)的返回值为 5。

12. 修正相角

名称：unwrap

修正相角。

语法：该函数有如下几种表达形式：

- Q = unwrap (P)
- Q = unwrap (P, tol)
- Q = unwrap (P, [], dim)
- Q = unwrap (P, tol, dim)

描述：当相邻数组元素之间的绝对跳跃值大于π，则函数 Q = unwrap (P)通过增加π的倍数来修正数组 P 中的相角。

Q = unwrap (P, tol)将绝对跳跃值π改为 tol 所指定的值。

Q = unwrap (P, [], dim)沿着 dim 指定的维数进行相角的修正。

Q = unwrap (P, tol, dim)沿着 dim 指定的维数进行相角的修正，同时使用 tol 所指定的值为绝对跳跃值。

举例：

对于如下所示的矩阵 P 中，除了元素(3，1)和(1，2)外，各个相角元素均光滑过度：

```
P =
         0    7.0686    1.5708    2.3562
    0.1963    0.9817    1.7671    2.5525
    6.6759    1.1781    1.9635    2.7489
    0.5890    1.3744    2.1598    2.9452
```

利用函数 Q = unwrap(P)可以消除这种元素之间的不连续性，结果为：

```
Q =
         0    0.7854    1.5708    2.3562
    0.1963    0.9817    1.7671    2.5525
    0.3927    1.1781    1.9635    2.7489
    0.5890    1.3744    2.1598    2.9452
```

7.6　向量函数

1. 向量的叉积

名称：cross

向量的叉积。

语法：该函数有如下两种表达形式：

- W = cross (U, V)
- W = cross (U, V, dim)

描述：

W = cross (U, V)返回向量 U 和 V 的叉积，即 W=U×V。

通常情况下，U 和 V 为包含 3 个元素的向量。

如果 U 和 V 是多维数组，则函数 W = cross (U, V, dim)返回 U 和 V 在 dim 指定的维数上的叉积。其中 U 和 V 必须有相同的尺寸，并且 size(U,dim)和 size(V,dim)必须均为 3。

举例：

对于向量 a = [1 2 3]和向量 b = [4 5 6]，利用函数 c = cross (a, b)可得二者之间的叉积，结果为：c = [-3　6　-3]。

对于两个任意的矩阵：

```
U =
       1       2       3
       4       5       6
       9       3       5
V =
      12      13      16
      30       6      39
       1       6      20
```

利用函数 W = cross (U, V)可得这两个矩阵列向量的叉积为：

```
W =
    -266      12     -75
     107      27      20
     -18     -53      21
```

利用函数 W = cross (U, V, 2)可得这两个矩阵行向量的叉积为：

```
W =
      -7      20     -11
     159      24    -126
      30    -175      51
```

2. 两个向量求交集

名称：intersect

两个向量求交集。

语法：该函数有如下几种表达形式：

- c = intersect(a,b)
- c = intersect(A,B,'rows')
- [c,ia,ib] = intersect(...)

描述：c = intersect(a,b)返回 a 和 b 中所共有的值。所得到的向量 c 中的元素按升序排列。

c = intersect(A,B,'rows')返回 A 和 B 中所共有的行向量，其中 A 和 B 为具有相同列数的矩阵。

[c,ia,ib] = intersect(...)同时还要返回列向量索引向量 ia 和 ib，并满足：

c = a(ia)和 c = b(ib)或 c= a(ia,:)和 c = b(ib,:)。

举例：

对于两个任意的向量 a = [1 2 3 6]和 b = [1 2 3 4 6 10 20]，利用函数 c = intersect(a,b)可得二者之间的交集，结果为：c =[1 2 3 6]。

对于两个任意的矩阵 A 和 B：

```
A =
     1     1     1
     3     6     8
     1     0     7
B =
     3     3     2
     3     6     8
     1     8     6
```

利用函数 c = intersect(A,B,'rows')可以求得 A 和 B 中所共有的行向量，结果如下所示：

c = 3 6 8。

3. 检验是否为集合中的元素

名称：ismember

检验是否为集合中的元素。

语法：该函数有如下两种表达形式：

- k = ismember(a,S)
- k = ismember(A,S,'rows')

描述：k = ismember(a,S)返回一个和向量 a 具有同样长度的向量，该向量中的元素描述了对应的 a 中各个元素的状态：取 1 表示 a 中对应的元素属于集合 S；取 0 表示 a 中对应的元素不属于集合 S。而且 a 和 S 可以是字符串。

k = ismember(A,S,'rows') 返回一个描述矩阵 A 中各行向量状态的向量。取 1 表示 A 中对应的行向量属于集合 S；取 0 表示 A 中对应的行向量不属于集合 S。

举例：

给定集合 S = [0 2 4 6 8 10 12 14 16 18 20]与任意一个向量 a =[1 2 3 4 5]。利用函数 k = ismember(a,S)可以得到描述 a 中各个元素状态的向量 k =[0 1 0 1 0]。

4. 两个向量的差集

名称：setdiff

求两个向量的差集。

语法：该函数有如下几种表达形式：

- c = setdiff(a,b)
- c = setdiff(A,B,'rows')
- [c,i] = setdiff(...)

描述：c = setdiff(a,b)返回 a 中有但 b 中没有的元素。结果向量 c 按升序进行排列，而且 a 和 b 也可以是字符串。

当 A 和 B 是含有相同列数的矩阵时，函数 c = setdiff(A,B,'rows')返回矩阵 A 中有但矩阵 B 中没有的行向量。

[c,i] = setdiff(...)同时还返回一个索引向量。

举例：

例如对于两个矩阵 A 和 B：

```
A =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
B =
     1     2     3     4     5
    23     5     7    14    16
     3     6     9    10    13
```

利用函数可以 c = setdiff(A,B,'rows')可以得到 A 和 B 中行向量的差集：

```
c =
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
    17    24     1     8    15
```

利用函数[c,i] = setdiff(A,B,'rows')还可以得到一个索引向量：

```
i =
     3
     4
     5
     1
```

5. 两个向量的异或

名称： setxor

求两个向量的异或。

语法： 该函数有如下几种表达形式：

- c = setxor(a,b)
- c = setxor(A,B,'rows')
- [c,ia,ib] = setxor(...)

描述： c = setxor(a,b)返回不在 a 和 b 交集中的元素。结果向量 c 按升序进行排列，而且 a 和 b 可以为字符串。

当 A 和 B 是含有相同列数的矩阵时，函数 c = setxor(A,B,'rows')返回不在 A 和 B 交集中的行向量。

[c,ia,ib] = setxor(...)同时还返回索引向量 ia 和 ib。

举例：

例如对于两个向量 a = [-1 0 1 Inf -Inf NaN]和 b = [-2 pi 0 Inf]，利用函数 c = setxor(a,b)可以求得二者之间的异或，结果为 c =[-Inf　-2.0000　-1.0000　1.0000　3.1416　NaN]。

6. 两个向量的并

名称： union

求两个向量的并。

语法：该函数有如下几种表达形式：

- c = union(a,b)
- c = union(A,B,'rows')
- [c,ia,ib] = union(...)

描述：c = union(a,b)返回属于 a 或 b 的元素(对于重复元素，只返回一个)。结果向量 c 按升序进行排列，而且 a 和 b 可以为字符串。

当 A 和 B 是含有相同列数的矩阵时，函数 c = union(A,B,'rows') 返回属于 A 或 B 的行向量(对于重复元素，只返回一个)。

[c,ia,ib] = union(...)同时还返回索引向量 ia 和 ib。

举例：

例如对于两个向量 a = [-1 0 2 4 6]和 b = [-1 0 1 3]，利用函数[c,ia,ib] = union(a,b)可以得到向量 a 和 b 的并，同时返回索引向量 ia 和 ib，结果如下所示：

```
c = -1    0    1    2    3    4    6
ia = 3    4    5
ib = 1    2    3    4
```

7. 向量的元素值

名称：unique

返回向量的元素值(无相同的值)。

语法：该函数有如下几种表达形式：

- b = unique(a)
- b = unique(A,'rows')
- [b,i,j] = unique(...)

描述：b = unique(a)返回向量 a 中的元素值，但不出现重复现象。结果向量 b 中的元素按升序排列，而且 a 可以是字符串。

b = unique(A,'rows')返回矩阵 A 中的行向量，也不出现重复现象。

[b,i,j] = unique(...)同时还返回索引向量 i 和 j。

举例：

例如对于向量 a = [1 1 5 6 2 3 3 9 8 6 2 4]，用函数[b,i,j] = unique(a)可以得到向量 a 中的元素(相同元素只出现一次)，同时返回索引向量 i 和 j，结果如下所示：

```
b = 1    2    3    4    5    6    8    9
i = 2    11   7    12   3    10   9    8
j = 1    1    5    6    2    3    3    8    7    6    2    4。
```

第八章　多项式和插值函数

8.1　多项式

1. 卷积和多项式相乘

名称：conv

可参见第十一章中的说明。

2. 解卷积和多项式相除

名称：deconv

可参见第十一章中的说明。

3. 多项式表达式

名称：poly

求已知根的多项式表达式。

语法：该函数有如下两种表达形式：

- p = poly(A)
- p = poly(r)

描述：p = poly(A)返回一个包含 n+1 个元素的行向量，该行向量中的元素是特征多项式 det(sI- A)的系数。其中 A 是一个 n×n 矩阵，系数按降幂顺序排列。如果向量 c 包含 n+1 个分量，则所表达的多项式为$c_1s^n+\dots+c_ns+c_{n+1}$。

p = poly(r)返回一个行向量，该行向量中的元素是以 r 中的元素为根的多项式系数。

举例：

对于如下所示的矩阵：

```
A =
     1     2     3
     4     5     6
     7     8     0
```

在 MATLAB 命令窗口中输入 p = poly(A)，可求得对应的多项式系数为：

```
p = 1.0000    -6.0000   -72.0000   -27.0000
```

4. 多项式求导

名称：polyder

多项式求导。

语法：该函数有如下几种表达形式：

- k = polyder(p)

- k = polyder(a,b)
- [q,d] = polyder(b,a)

描述：函数 polyder 可以用来计算多项式及其积和商的导数。

k = polyder(p)返回多项式 p 的导数。

k = polyder(a,b) 返回多项式 a 和 b 积的导数。

[q,d] = polyder(b,a)返回多项式商 b/a 的导数的分子 q 和分母 d。

举例：

本例求解两个多项式的积$(3x^2+6x+9)(x^2+2x)$的导数。

这两个多项式对应的系数分别为 a = [3 6 9]和 b = [1 2 0]。

利用函数 k = polyder(a,b)可以求得这两个多项式乘积的导数的系数，结果为：

k =[12 36 42 18]，即得到的多项式为 $12x^3+36x^2+42x+18$。

5. 特征值问题

名称：polyeig

多项式的特征值问题。

语法：[X,e] = polyeig(A0,A1,...Ap)

描述：函数[X,e] = polyeig(A0,A1,...Ap)求解如下所示的 p 阶多项式特征值问题：

$(A_0+\lambda A_1+...+\lambda^p A_p)x=0$。

其中 p 为非负的整数，A_0、A_1,... A_p 为 n 阶输入矩阵。

输出矩阵 X(n×n*p 阶)的列向量就是多项式的特征向量。

输出向量 e 中的元素就是多项式的特征值。

基于不同的 p 和 n 值，函数 polyeig 可以得到几种不同的结果：

当 p=0 时，polyeig(A)就蜕化为标准特征值问题：eig(A)。

当 p= 1 时，polyeig(A, B)就变为广义特征值问题：eig(A, -B)。

当 n= 1 ，并且 a0,a1,...ap 为标量时，polyeig(a0,a1,...ap)就变为标准多项式问题：roots([ap ... a1 a0])。

举例：

对于如下所示的四个矩阵 A0、A1、A2 和 A3：

```
A0 =
     1     1     1
     2     4     7
     6     9     1
A1 =
     1     2     3
     4     5     6
     3     7     9
A2 =
    11    21    33
    46    51    16
    30    17    24
A3 =
```

```
    23     6    13
    19    31     6
    25     9    14
```

利用函数[X,e] = polyeig(A0,A1,A2,A3)可以求得相应的特征向量和特征值：

```
X =
  Columns 1 through 4
  -0.0001 + 0.0000i    0.0696 - 0.0000i   -0.1863 - 0.2236i   -0.2238 - 0.1861i
   0.0000 + 0.0000i    0.2772 - 0.0000i    0.1664 + 0.4460i    0.4462 + 0.1660i
   0.0002 - 0.0000i   -0.0953 + 0.0000i   -0.0130 - 0.4372i   -0.4372 - 0.0126i
  Columns 5 through 8
   0.8309 - 0.0545i   -0.0456 - 0.8314i    0.0644 + 0.7585i   -0.7451 - 0.1558i
  -0.3833 + 0.0530i   -0.0066 + 0.3868i   -0.1549 - 0.5517i    0.5289 + 0.2206i
   0.2356 + 0.1182i   -0.1456 - 0.2197i    0.0613 - 0.0131i    0.0204 - 0.0593i
  Column 9
  -0.7463 + 0.0000i
   0.6107 + 0.0000i
  -0.1058 + 0.0000i
e =
  67.8593 - 0.0000i
  -1.6443 + 0.0000i
  -0.6700 - 0.4054i
  -0.6700 + 0.4054i
   0.0037 - 0.2969i
   0.0037 + 0.2969i
   0.2741 - 0.1172i
   0.2741 + 0.1172i
  -0.2430
```

6.　曲线拟合

名称：polyfit

多项式的曲线拟合。

语法：该函数有如下两种表达形式：

- p = polyfit(x,y,n)
- [p,s] = polyfit(x,y,n)

描述：对于实验或统计数据，为了描述不同变量之间的关系，经常采用拟合曲线的办法。拟合曲线，就是要根据已知数据找出相应函数的系数。通常情况下，已知数据往往多于未知系数的个数，所以曲线拟合实质上是解超线性方程组。

函数 polyfit 从最小二乘的意义上，拟合出所给数据的多项式系数。

p = polyfit(x,y,n)在向量 p 中返回多项式系数。其中 x 和 y 为已知数据的横坐标和纵坐标向量，n 为多项式的次数。

[p.s] = polyfit(x,y,n)同时还返回一个误差估计数组 s。

举例：

本例中对一组给定的点进行拟合。

首先利用函数 x = (0: 0.1: 2.5)'生成一个在区间[0, 2.5]内等间距分布的 x 点集向量，并利用函数 y = erf(x)估计这些点的误差，如下所示：

x =	y=
0	0
0.1000	0.1125
0.2000	0.2227
0.3000	0.3286
0.4000	0.4284
0.5000	0.5205
0.6000	0.6039
0.7000	0.6778
0.8000	0.7421
0.9000	0.7969
1.0000	0.8427
1.1000	0.8802
1.2000	0.9103
1.3000	0.9340
1.4000	0.9523
1.5000	0.9661
1.6000	0.9763
1.7000	0.9838
1.8000	0.9891
1.9000	0.9928
2.0000	0.9953
2.1000	0.9970
2.2000	0.9981
2.3000	0.9989
2.4000	0.9993
2.5000	0.9996

然后利用函数 p = polyfit(x,y,6)得到 6 阶近似多项式系数：

p = 0.0084　-0.0983　0.4217　-0.7435　0.1471　1.1064　0.0004。

对应的多项式为：

$0.0084x^6-0.0983x^5+0.4217x^4-0.7435x^3+0.1471x^2+1.1064x+0.0004$。

为了观察拟合的效果，使用函数 f = polyval(p,x)来进行评价，并利用函数 table = [x y f y-f] 将数据、拟合结果和误差在表中列出来：

table =

```
         0         0    0.0004   -0.0004
    0.1000    0.1125    0.1119    0.0006
    0.2000    0.2227    0.2223    0.0004
    0.3000    0.3286    0.3287   -0.0001
    0.4000    0.4284    0.4288   -0.0004
    0.5000    0.5205    0.5209   -0.0004
    0.6000    0.6039    0.6041   -0.0002
    0.7000    0.6778    0.6778    0.0000
    0.8000    0.7421    0.7418    0.0003
    0.9000    0.7969    0.7965    0.0004
    1.0000    0.8427    0.8424    0.0003
    1.1000    0.8802    0.8800    0.0002
    1.2000    0.9103    0.9104   -0.0000
    1.3000    0.9340    0.9342   -0.0002
    1.4000    0.9523    0.9526   -0.0003
    1.5000    0.9661    0.9664   -0.0003
    1.6000    0.9763    0.9765   -0.0002
    1.7000    0.9838    0.9838    0.0000
    1.8000    0.9891    0.9889    0.0002
    1.9000    0.9928    0.9925    0.0003
    2.0000    0.9953    0.9951    0.0002
    2.1000    0.9970    0.9969    0.0001
    2.2000    0.9981    0.9982   -0.0001
    2.3000    0.9989    0.9991   -0.0003
    2.4000    0.9993    0.9995   -0.0002
    2.5000    0.9996    0.9994    0.0002
```

为了更直观，调用函数 plot(x,y,'o',x,f,'-')，将拟合后的曲线在窗口中显示出来，如图 8-1 所示。

7. 多项式求值

名称：polyval

多项式求值。

语法：该函数有如下两种表达形式：

- y = polyval(p,x)
- [y,delta] = polyval(p,x,S)

描述：y = polyval(p,x)返回多项式 p 在 x 点处的取值。x 可以是向量，也可以是矩阵。[y,delta] = polyval(p,x,S)同时还生成误差估计。

举例：

本例求多项式 $p(x)=3x^2+2x+1$ 在 x=5, 7 和 9 处的取值。

其中系数向量 p = [3 2 1]，利用函数 y = polyval(p,[5 7 9])可求得对应的值：

y = 86，162 和 262。

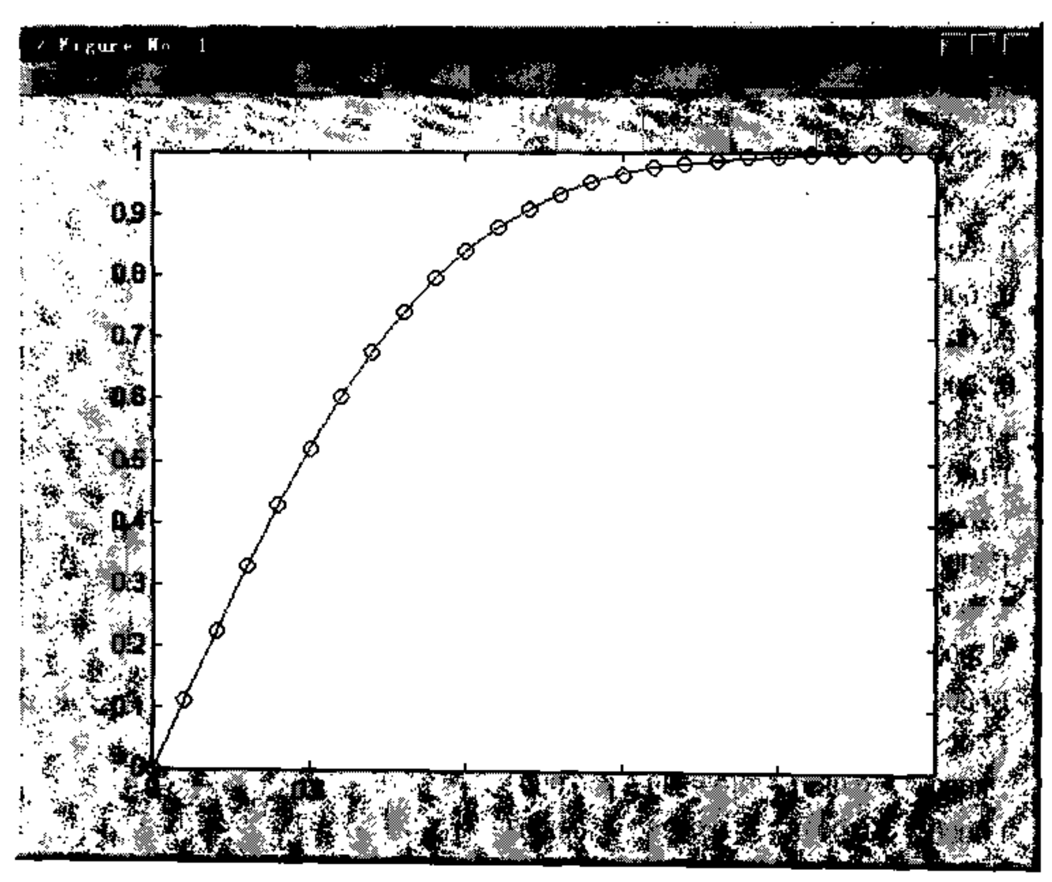


图 8-1 拟合曲线

8. 矩阵多项式的值

名称：polyvalm

求矩阵多项式的值。

语法：Y = polyvalm(p,X)

描述：Y = polyvalm(p,X)返回矩阵意义上的多项式的值。其中 p 为向量，其元素为按降序排列的多项式的系数，而 X 必须是方阵。

举例：

对于 4 阶 Pascal 矩阵：

```
X =
     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
```

利用函数 p = poly(X)可以生成其特征多项式：

p = 1.0000 -29.0000 72.0000 -29.0000 1.0000。

对应的多项式为：$x^4-29x^3+72x^2-29x+1$。

首先利用函数 polyval(p,X)计算该多项式在每个元素处的取值，可得：

```
ans =
  1.0e+004 *
    0.0016    0.0016    0.0016    0.0016
    0.0016    0.0015   -0.0140   -0.0563
    0.0016   -0.0140   -0.2549   -1.2089
    0.0016   -0.0563   -1.2089   -4.3779
```

再利用函数 polyvalm(p,X)计算矩阵多项式的取值，可得：

```
ans =
  1.0e-011 *
   -0.0077    0.0053   -0.0096    0.0430
   -0.0068    0.0481   -0.0110    0.1222
    0.0075    0.1400   -0.0095    0.2608
    0.0430    0.2920   -0.0007    0.4737
```

结果为一个零矩阵(存在计算机的数值误差)。这恰好是 Cayley-Hamilton 理论的一个例证：矩阵满足其自身的特征方程。

9. 两个多项式的比值

名称：residue

展开两个多项式的比值。

语法：该函数有如下两种表达形式：

- [r,p,k] = residue(b,a)
- [b,a] = residue(r,p,k)

描述：这是一个对系统转移函数特别有用的函数。

[r,p,k] = residue(b,a)返回两个多项式 b(s)和 a(s)之比的部分分式展开项中的残差、极和直接项。用代数表达为：

$$\frac{b(s)}{a(s)}=\frac{b_1+b_2s^{-1}+b_3s^{-2}+...+b_{m+1}s^{-m}}{a_1+a_2s^{-1}+a_3s^{-2}+...+a_{n+1}s^{-n}}。$$

[b,a] = residue(r,p,k)将部分分式展开项还原成多项式表达式。

举例：

对于由 a=[1 2 3 4]和 b=[3 6 8 1]为系数所描述的两个多项式，利用函数[r,p,k] = residue(b,a)可以求得这两个多项式之比的部分分式展开项中的残差、极和直接项，结果如下所示：

```
r =
  -2.0453
    1.0226 + 1.2990i
    1.0226 - 1.2990i
p =
  -1.6506
  -0.1747 + 1.5469i
  -0.1747 - 1.5469i
k = 3
```

10. 多项式求根

名称：roots

多项式求根。

语法：r = roots(c)

描述：r = roots(c)返回一个元素为多项式 c 的根的列向量。行向量 c 中包含按降幂排列的多项式的系数，如果 c 中包含 n+1 个元素，则多项式的表达式为：$c_1s^n+\ldots c_ns+c_{n+1}$。

举例：

例如对于多项式 $s^3-6s^2-72s-27$，其系数向量为：p = [1 -6 -72 -27]。利用函数 r = roots(p) 可以得到该多项式的根：

```
r =
    12.1229
    -5.7345
    -0.3884
```

8.2 数据插值

1. 三维网格点数据

名称：griddata

求解三维网格点数据。

语法：该函数有如下几种表达形式：

- ZI = griddata(x,y,z,XI,YI)
- [XI,YI,ZI] = griddata(x,y,z,xi,yi)
- [...] = griddata(...,method)

描述：ZI = griddata(x,y,z,XI,YI)返回与向量 x、y 和 z 所描述的数据点集相匹配的表面 f(x,y)上网格点的 z 坐标矩阵 ZI。函数 griddata 在点(XI, YI)处对表面函数 f(x,y)进行插值，从而得到 ZI 的值。XI 和 YI 通常形成均匀网格。

[XI,YI,ZI] = griddata(x,y,z,xi,yi)除了返回与上面相同的 ZI 矩阵，同时还要返回由行向量 xi 和列向量 yi 组成的 XI 和 YI 矩阵。

[...] = griddata(...,method)采用指定的插值方法。

当 method 取‘linear’时，采用基于三角形的线性插值方法(缺省)。

当 method 取‘cubic’时，采用基于三角形的三次插值方法。

当 method 取‘nearest’时，采用最邻近插值方法。

当 method 取‘v4’时，采用 MATLAB 中的 griddata 法。

举例：

本例中使用在-2.0 和 2.0 之间随机产生的 100 数据点来绘制三维网格图。

首先调用函数 rand('seed',0)，设置产生随机数的方式。接着调用函数 x = rand(100,1)*4-2，y = rand(100,1)*4-2 和 z = x.*exp(-x.^2-y.^2)生成数据点的坐标值：

```
x =                    y =                   z =
```

-1.1242	0.2866	-0.2926
-1.8118	1.2096	-0.0157
0.7155	-1.8678	0.0131
0.7172	0.1378	0.4207
1.7388	-0.0061	0.0846
-0.4660	1.8214	-0.0136
0.0777	0.9932	0.0288
1.3239	0.2183	0.2188
-1.8617	1.5629	-0.0051
-1.7862	0.4994	-0.0573
0.1188	1.3682	0.0180
0.6846	-1.3609	0.0672
-1.9692	-1.1490	-0.0109
-0.4663	0.8588	-0.1794
-1.7326	-1.4783	-0.0097
-0.3301	-1.6360	-0.0204
0.7471	-0.9016	0.1896
0.3559	-1.9880	0.0060
1.7217	-0.3428	0.0790
1.3847	-1.8925	0.0057
0.1077	0.8393	0.0526
-1.6321	1.7516	-0.0053
0.6157	-1.0404	0.1428
-0.3360	-1.2764	-0.0588
0.8048	-0.7298	0.2472
1.6413	1.5480	0.0101
1.0488	0.6082	0.2412
-0.9502	-1.3987	-0.0545
-1.8101	0.7254	-0.0404
0.9443	-0.4567	0.3142
-0.6871	-0.4491	-0.3503
0.5306	-0.0010	0.4004
1.0256	-1.4099	0.0491
1.9641	0.3487	0.0367
-0.5386	1.3823	-0.0596
-1.0118	0.3604	-0.3192
1.9302	1.8216	0.0017
0.8906	0.2246	0.3831
1.0134	-1.4074	0.0501

```
    0.6061    1.9332    0.0100
   -1.7093   -0.3649   -0.0806
    0.5265   -1.4327    0.0512
    1.5388    0.2596    0.1347
   -0.9092   -0.9915   -0.1488
   -0.2544   -0.0459   -0.2379
    1.0660   -0.1439    0.3352
   -0.0891    1.8444   -0.0029
   -1.0489   -1.4959   -0.0373
   -0.9004   -1.2010   -0.0946
   -0.5629   -0.7230   -0.2431
   -1.3340    0.5171   -0.1723
   -0.0539   -1.4932   -0.0058
    1.5906    0.6050    0.0879
    1.6368    0.4865    0.0886
   -1.7577    1.2123   -0.0184
    1.6186   -1.0086    0.0426
    0.0181   -0.0943    0.0179
    0.0652   -0.4427    0.0533
   -0.7239   -1.1870   -0.1048
    1.9466   -1.8865    0.0013
   -0.0241    1.6067   -0.0018
   -0.9354   -0.2940   -0.3576
   -1.6371   -1.4319   -0.0144
    1.7911    1.7899    0.0029
   -1.7050   -0.3587   -0.0819
    0.0028   -1.4752    0.0003
   -0.4634    1.5426   -0.0346
   -0.8917   -1.6313   -0.0281
    1.6553   -1.3512    0.0172
    0.1190   -1.7157    0.0062
   -0.1422   -0.5386   -0.1043
    1.7639   -0.9878    0.0296
   -1.7997   -1.4596   -0.0084
    1.0461    1.1326    0.0971
    1.0808   -0.1788    0.3255
    1.3113   -0.6019    0.1635
   -1.4985   -0.1908   -0.1530
   -1.9365    1.2358   -0.0099
```

```
   0.7538          1.7267          0.0217
   1.4730          0.6066          0.1164
   0.5182         -1.1390          0.1083
   0.9449          0.7184          0.2309
   0.9016          1.6357          0.0275
   1.9978         -0.9995          0.0136
   1.5543          1.4434          0.0173
  -1.0672         -0.1150         -0.3372
  -0.7747          0.0238         -0.4249
  -0.5959          0.4016         -0.3556
   0.0531          1.2702          0.0105
   0.3645          1.0234          0.1120
   1.3839         -0.1510          0.1993
  -0.3517          1.8055         -0.0119
   1.3660          0.5310          0.1594
  -0.9227         -0.2427         -0.3713
  -0.3384          1.2988         -0.0559
   0.1492          0.7559          0.0824
  -0.1283          0.8088         -0.0656
  -0.8512          1.9486         -0.0093
  -1.2867          1.8177         -0.0090
  -1.3851          1.4051         -0.0282
```

接下来，为这些数据点定义一个规则网格。首先利用函数 ti = -2:.25:2 产生等间距的数组：

```
ti =
  Columns 1 through 7
   -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  Columns 8 through 14
   -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  Columns 15 through 17
    1.5000    1.7500    2.0000
```

然后调用函数[XI,YI] = meshgrid(ti,ti)产生网格点的 x 和 y 坐标值：

```
XI =
  Columns 1 through 7
   -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
   -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
   -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
   -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
   -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
```

```
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
  -2.0000   -1.7500   -1.5000   -1.2500   -1.0000   -0.7500   -0.5000
 Columns 8 through 14
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
  -0.2500         0    0.2500    0.5000    0.7500    1.0000    1.2500
 Columns 15 through 17
   1.5000    1.7500    2.0000
   1.5000    1.7500    2.0000
   1.5000    1.7500    2.0000
   1.5000    1.7500    2.0000
   1.5000    1.7500    2.0000
   1.5000    1.7500    2.0000
   1.5000    1.7500    2.0000
   1.5000    1.7500    2.0000
```

```
    1.5000    1.7500    2.0000
    1.5000    1.7500    2.0000
    1.5000    1.7500    2.0000
    1.5000    1.7500    2.0000
    1.5000    1.7500    2.0000
    1.5000    1.7500    2.0000
    1.5000    1.7500    2.0000
    1.5000    1.7500    2.0000
    1.5000    1.7500    2.0000
YI =
  Columns 1 through 7
   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000
   -1.7500   -1.7500   -1.7500   -1.7500   -1.7500   -1.7500   -1.7500
   -1.5000   -1.5000   -1.5000   -1.5000   -1.5000   -1.5000   -1.5000
   -1.2500   -1.2500   -1.2500   -1.2500   -1.2500   -1.2500   -1.2500
   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000
   -0.7500   -0.7500   -0.7500   -0.7500   -0.7500   -0.7500   -0.7500
   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000
   -0.2500   -0.2500   -0.2500   -0.2500   -0.2500   -0.2500   -0.2500
         0         0         0         0         0         0         0
    0.2500    0.2500    0.2500    0.2500    0.2500    0.2500    0.2500
    0.5000    0.5000    0.5000    0.5000    0.5000    0.5000    0.5000
    0.7500    0.7500    0.7500    0.7500    0.7500    0.7500    0.7500
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
    1.2500    1.2500    1.2500    1.2500    1.2500    1.2500    1.2500
    1.5000    1.5000    1.5000    1.5000    1.5000    1.5000    1.5000
    1.7500    1.7500    1.7500    1.7500    1.7500    1.7500    1.7500
    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000
  Columns 8 through 14
   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000
   -1.7500   -1.7500   -1.7500   -1.7500   -1.7500   -1.7500   -1.7500
   -1.5000   -1.5000   -1.5000   -1.5000   -1.5000   -1.5000   -1.5000
   -1.2500   -1.2500   -1.2500   -1.2500   -1.2500   -1.2500   -1.2500
   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000
   -0.7500   -0.7500   -0.7500   -0.7500   -0.7500   -0.7500   -0.7500
   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000
   -0.2500   -0.2500   -0.2500   -0.2500   -0.2500   -0.2500   -0.2500
         0         0         0         0         0         0         0
    0.2500    0.2500    0.2500    0.2500    0.2500    0.2500    0.2500
```

```
    0.5000    0.5000    0.5000    0.5000    0.5000    0.5000    0.5000
    0.7500    0.7500    0.7500    0.7500    0.7500    0.7500    0.7500
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
    1.2500    1.2500    1.2500    1.2500    1.2500    1.2500    1.2500
    1.5000    1.5000    1.5000    1.5000    1.5000    1.5000    1.5000
    1.7500    1.7500    1.7500    1.7500    1.7500    1.7500    1.7500
    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000
  Columns 15 through 17
   -2.0000   -2.0000   -2.0000
   -1.7500   -1.7500   -1.7500
   -1.5000   -1.5000   -1.5000
   -1.2500   -1.2500   -1.2500
   -1.0000   -1.0000   -1.0000
   -0.7500   -0.7500   -0.7500
   -0.5000   -0.5000   -0.5000
   -0.2500   -0.2500   -0.2500
         0         0         0
    0.2500    0.2500    0.2500
    0.5000    0.5000    0.5000
    0.7500    0.7500    0.7500
    1.0000    1.0000    1.0000
    1.2500    1.2500    1.2500
    1.5000    1.5000    1.5000
    1.7500    1.7500    1.7500
    2.0000    2.0000    2.0000
```

再利用函数 ZI = griddata(x,y,z,XI,YI)产生相应的 z 坐标值：

```
ZI =
  Columns 1 through 7
       NaN       NaN       NaN       NaN       NaN       NaN       NaN
       NaN       NaN       NaN       NaN       NaN       NaN   -0.0142
       NaN       NaN   -0.0172   -0.0280   -0.0392   -0.0496   -0.0466
       NaN   -0.0235   -0.0516   -0.0723   -0.0808   -0.0937   -0.0771
       NaN   -0.0450   -0.0758   -0.1065   -0.1373   -0.1554   -0.1532
       NaN   -0.0570   -0.0878   -0.1185   -0.1942   -0.2371   -0.2132
       NaN   -0.0691   -0.1074   -0.1898   -0.2721   -0.3246   -0.2617
       NaN   -0.0691   -0.1536   -0.2436   -0.3399   -0.3801   -0.3040
       NaN   -0.0685   -0.1465   -0.2519   -0.3520   -0.4141   -0.3260
       NaN   -0.0687   -0.1384   -0.2352   -0.3339   -0.3743   -0.3229
       NaN   -0.0665   -0.1304   -0.1985   -0.2507   -0.2942   -0.2919
```

```
      NaN   -0.0534   -0.1148   -0.1414   -0.1590   -0.1766   -0.2210
      NaN   -0.0355   -0.0791   -0.1018   -0.1195   -0.1371   -0.1467
      NaN   -0.0177   -0.0401   -0.0623   -0.0750   -0.0828   -0.0874
      NaN   -0.0112   -0.0203   -0.0264   -0.0352   -0.0440   -0.0424
      NaN       NaN   -0.0081   -0.0130   -0.0178   -0.0266   -0.0195
      NaN       NaN       NaN       NaN       NaN       NaN       NaN
  Columns 8 through 14
      NaN       NaN       NaN       NaN       NaN       NaN       NaN
  -0.0137   -0.0043    0.0131    0.0248    0.0241    0.0207    0.0187
  -0.0261   -0.0006    0.0234    0.0454    0.0502    0.0417    0.0323
  -0.0435    0.0061    0.0539    0.0850    0.0961    0.0840    0.0482
  -0.0730   -0.0193    0.0343    0.1162    0.1632    0.1327    0.0960
  -0.1123   -0.0488    0.0683    0.1389    0.2238    0.1853    0.1439
  -0.1566   -0.0049    0.1070    0.2077    0.2968    0.2833    0.2044
  -0.2004    0.0040    0.1912    0.2981    0.3531    0.3304    0.2422
  -0.2299   -0.0052    0.1889    0.3773    0.3873    0.3529    0.2540
  -0.1983   -0.0220    0.1788    0.3243    0.3822    0.3359    0.2403
  -0.1667   -0.0312    0.1687    0.2419    0.2840    0.2821    0.1982
  -0.1280   -0.0013    0.1017    0.1475    0.1937    0.2115    0.1426
  -0.0762    0.0051    0.0799    0.1239    0.1317    0.1395    0.0933
  -0.0387    0.0028    0.0556    0.0826    0.0808    0.0809    0.0659
  -0.0273    0.0020    0.0166    0.0362    0.0472    0.0426    0.0272
  -0.0088   -0.0013    0.0033    0.0140    0.0205    0.0177    0.0139
      NaN       NaN       NaN       NaN       NaN       NaN       NaN
  Columns 15 through 17
      NaN       NaN       NaN
   0.0083    0.0063       NaN
   0.0185    0.0124       NaN
   0.0329    0.0164       NaN
   0.0609    0.0302       NaN
   0.0990    0.0477       NaN
   0.1230    0.0658       NaN
   0.1538    0.0744       NaN
   0.1574    0.0818       NaN
   0.1499    0.0854       NaN
   0.1252    0.0657       NaN
   0.0963    0.0476       NaN
   0.0708    0.0295       NaN
   0.0434    0.0165       NaN
```

```
    0.0163    0.0106       NaN
    0.0102    0.0045       NaN
       NaN       NaN       NaN
```

最后调用 mesh(XI,YI,ZI)函数，在窗口中绘制该规则网格图，如图 8-2 所示。

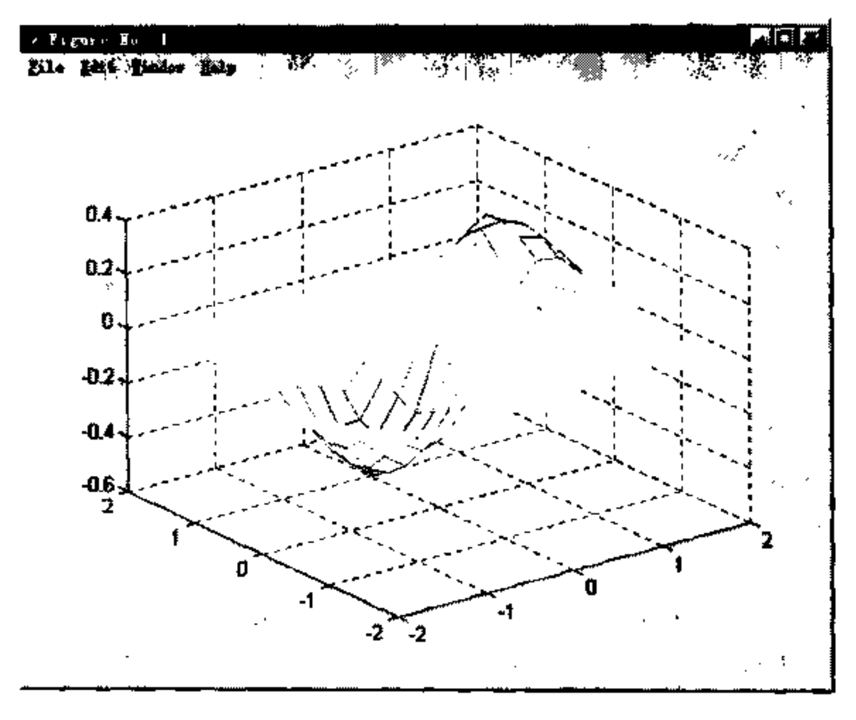


图 8-2 规则网格图

为了比较起见，利用函数 plot3(x,y,z,'o')将数据点在窗口中显示出来，如图 8-3 所示。

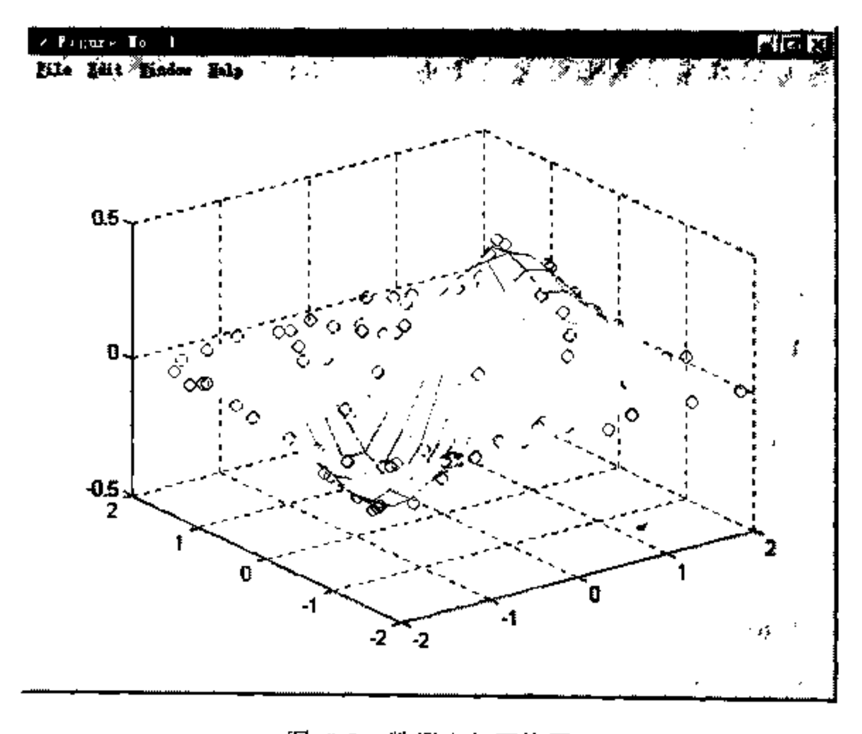


图 8-3 数据点与网格图

2. 一维插值

名称：interp1

一维插值。

语法：该函数有如下两种表达形式：

- yi = interp1(x,Y,xi)
- yi = interp1(x,Y,xi,method)

描述：在已知的数据点之间利用某种算法估算出新的数据点，称为插值。插值在信号和图像处理方面有很重要的应用。MATLAB 提供了好几种插值方法来平滑已知数据，包括一维插值、二维插值和高维插值。此处先介绍一维插值方法。

在 MATLAB 中有两类一维数据的插值方法：多项式插值法和基于 FFT(快速傅里叶变换)的插值法。函数 interp1 采用多项式插值法，它用多项式拟合所给数据，然后在插值的点上，根据多项式算出相应的值。它的调用格式为：

yi = interp1(x,Y,xi)

yi = interp1(x,Y,xi,method)

其中，xi 为需要插值的位置所组成的向量，yi 为根据插值算法求得的值所组成的向量，x 和 Y 为已知的数据点向量。参量 method 用于确定具体的插值方法，包括：

取‘linear’时，表示采用线性插值方法。

取‘cubic’时，表示采用三次插值方法。

取‘nearest’时，表示采用最近点插值方法。

取‘spline’时，表示采用三次样条插值方法。

这四种方法都要求把已给数据按 x 做升序或降序排列。

其中线性插值法把相邻的数据点用直线连接，按所生成的曲线插值。最近点插值法根据已知两点间插值点和这两点间的位置的远近来插值，当插值点离前点较近时，插值点取前点的值；相反，则取后点的值。样条插值法利用已知数据求出样条函数后，按照样条函数插值。三次插值法根据已知数据拟合出立方函数来插值。除样条函数外，其余三种都只能用于内插。

在选择插值方法时，应该考虑速度、内存需要和光滑问题。在上述四种方法中，最近点插值法最快，但它的插值很粗糙。线性插值法较最近点插值法需要更多的内存和计算时间，但插值曲线连续，并且导数连续。样条插值法虽然比三次插值法所需的内存少，但耗时多，不过插值曲线最光滑。需要说明的是，由于样条插值的特性，当已知数据分布不均匀时，插值结果不太理想。

举例：

下面的两个向量分别包括了 1900 到 1990 年间美国人口普查的年代和相应的人口数(单位为百万)：

```
t =
  Columns 1 through 6
        1900        1910        1920        1930        1940        1950
  Columns 7 through 10
        1960        1970        1980        1990
```

```
p =
  Columns 1 through 7
   75.9950    91.9720   105.7110   123.2030   131.6690   150.6970   179.3230
  Columns 8 through 10
  203.2120   226.5050   249.6330
```

利用函数 interp1(t,p,1975)进行插值，从而估计 1975 年的人口数，结果为(单位为百万)：ans =214.8585。

现在估计一下 1990 到 2000 年每一年的人口数，利用 x = 1900:1:2000 生成相应的年代数，并利用函数 y = interp1(t,p,x,'spline')进行插值处理，进而利用函数 plot(t,p,'o',x,y)将结果在窗口中显示出来，如图 8-4 所示。

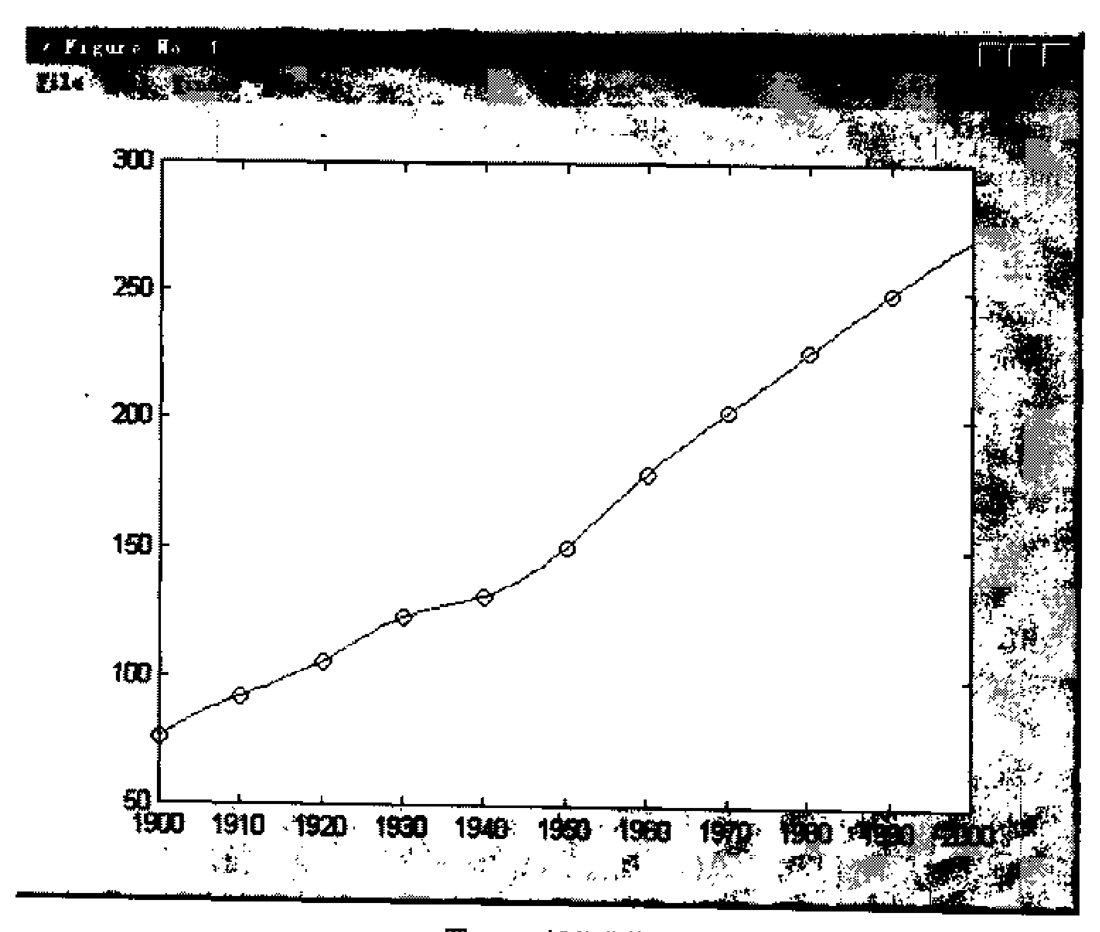


图 8-4 插值曲线

3. 二维插值

名称：interp2

二维插值。

- **语法**：该函数有如下几种表达形式：
- ZI = interp2(X,Y,Z,XI,YI)
- ZI = interp2(Z,XI,YI)
- ZI = interp2(Z,ntimes)
- ZI = interp2(X,Y,Z,XI,YI,method)

描述：前面已经介绍了 MATLAB 中一维插值函数的特点。此处介绍二维插值法。

二维插值即对二维数据进行插值。在图像处理和数据可视化方面，二维插值十分重要。其调用格式为：

ZI = interp2(X,Y,Z,XI,YI)

ZI = interp2(Z,XI,YI)

ZI = interp2(Z,ntimes)

ZI = interp2(X,Y,Z,XI,YI,method)

其中 Z 是由已知点的值组成的矩阵。X、Y 是和 Z 同维的已知点的 x 坐标矩阵和 y 坐标矩阵。XI 和 YI 为需要插值的点的 x 坐标阵和 y 坐标阵。ZI 将给出插值结果。参数 method 可以用来选择具体的插值方法，包括：

取 'linear' 时，表示采用双线性插值方法(缺省)。

取 'cubic' 时，表示采用双立方插值方法。

取 'nearest' 时，表示采用最近点插值方法。

取 'spline' 时，表示采用三次样条插值方法。

这四种方法都要求把已给数据按 x 做升序或降序排列。

和一维插值类似，最近点插值法把离插值点最近的已知点的值赋给插值点。双线性插值法利用已知点的值拟合出一个双线性曲面，然后根据插值点的坐标插值，该方法利用离每个插值点最近的四个点来近似给出该点的值。双立方插值法利用已知点的值拟合出一个双立方曲面(由许多双立方曲面块组成的曲面)，然后根据插值点的坐标插值。每个插值点的值由离插值点最近的 6 个已知点的值来近似，插值点处值和导数都连续。该方法插值效果较双线性插值法好，在图像处理中被广泛使用。

样条插值法利用已知数据求出样条函数后，按照样条函数插值。三次插值法根据已知数据拟合出立方函数来插值。除样条函数外，其余三种都只能用于内插。

另外，在二维插值时，MATLAB 会自动根据输入数据在坐标平面上生成等间距网格，如果在指定方法时加上星号，将提高插值速度。

举例：

首先利用函数[X,Y] = meshgrid(-3:.25:3)准备输入数据矩阵，然后利用函数 Z = peaks(X,Y)得到数据点。

在 MATLAB 中输入[XI,YI] = meshgrid(-3:.125:3)，得到需要插值的点的 x 坐标阵和 y 坐标阵。并利用函数 ZI = interp2(X,Y,Z,XI,YI)得到插值结果。

为了直观起见，利用函数 mesh(X,Y,Z)将已知点组成的网格图显示出来。同时输入函数 hold 和 mesh(XI,YI,ZI+15)将得到的插值点所构成的网格图显示出来，如图 8-5 所示。

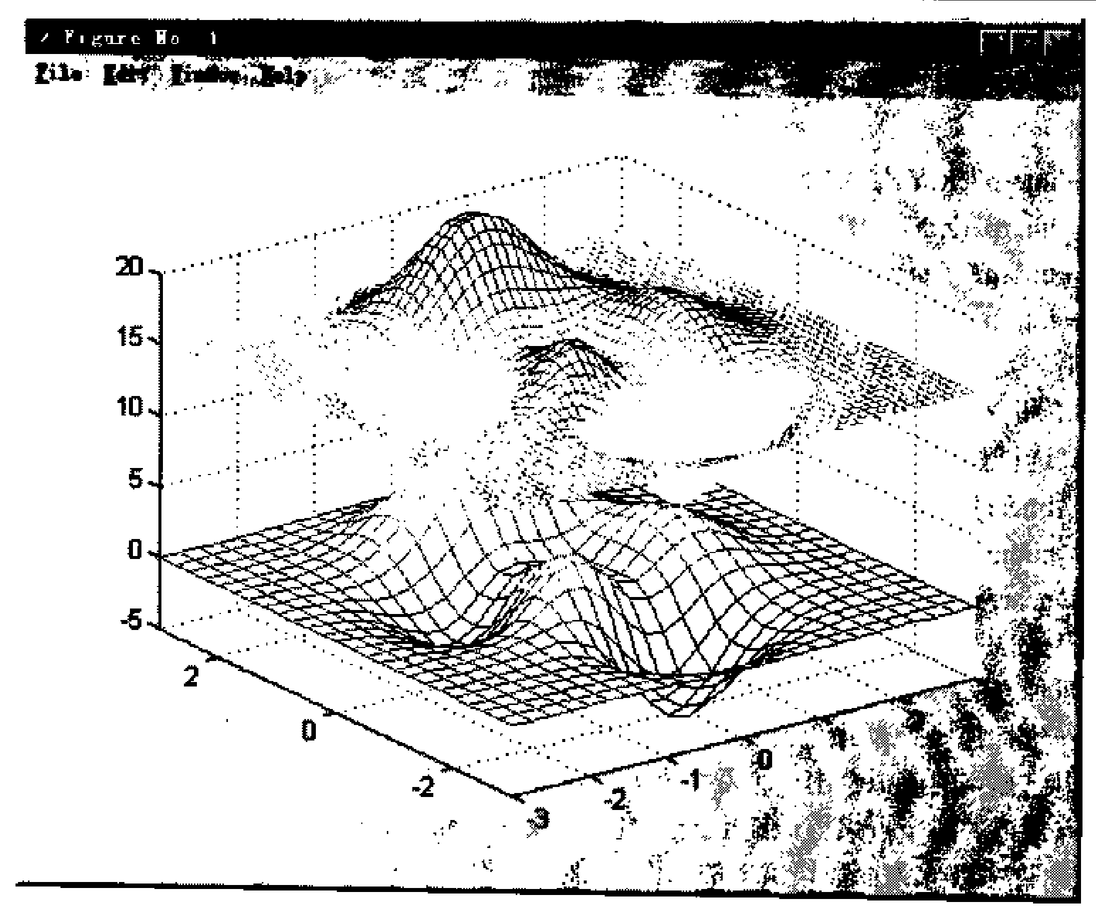


图 8-5　二维插值得到的网格图

4. 三维插值

名称：interp3

三维插值。

语法：该函数有如下几种表达形式：

- VI = interp3(X,Y,Z,V,XI,YI,ZI)
- VI = interp3(V,XI,YI,ZI)
- VI = interp3(V,ntimes)
- VI = interp3(...,method)

描述：三维插值即对三维数据进行插值。其调用格式为：

VI = interp3(X,Y,Z,V,XI,YI,ZI)

VI = interp3(V,XI,YI,ZI)

VI = interp3(V,ntimes)

VI = interp3(...,method)

其中 X、Y 和 Z 分别是已知数据点的 x 坐标矩阵、y 坐标矩阵和 z 坐标矩阵；XI、YI 和 ZI 分别为需要插值的点的 x 坐标矩阵、y 坐标矩阵和 z 坐标矩阵；V 为已知点的值；VI 为待插值点的值；参数 method 可以用来选择具体的插值方法，包括：

取 'linear' 时，表示采用三线性插值方法(缺省)。

取 'cubic' 时，表示采用三立方插值方法。

取 'nearest' 时，表示采用最近点插值方法。

取 'spline' 时，表示采用三次样条插值方法。

这四种方法都要求把已给数据按 x 做升序或降序排列。

其中，最近点插值法把空间位置上离插值点最近的已知点的值赋给插值点。三线性插值法把已知点用分块三线性曲面连接，再对插值点插值。三立方插值法将已知点用三立方曲面连接后再插值。

举例：

首先利用函数 [x,y,z,v] = flow(10) 得到已知点的数据。然后利用函数 [xi,yi,zi]=meshgrid(.1:.25:10, -3:.25:3, -3:.25:3)得到插值点的数据。

在 MATLAB 中输入 vi = interp3(x,y,z,v,xi,yi,zi)，得到插值结果。

最后调用函数 slice(xi,yi,zi,vi,[6 9.5],2,[-2 .2])，将结果在窗口中显示出来，如图 8-6 所示。

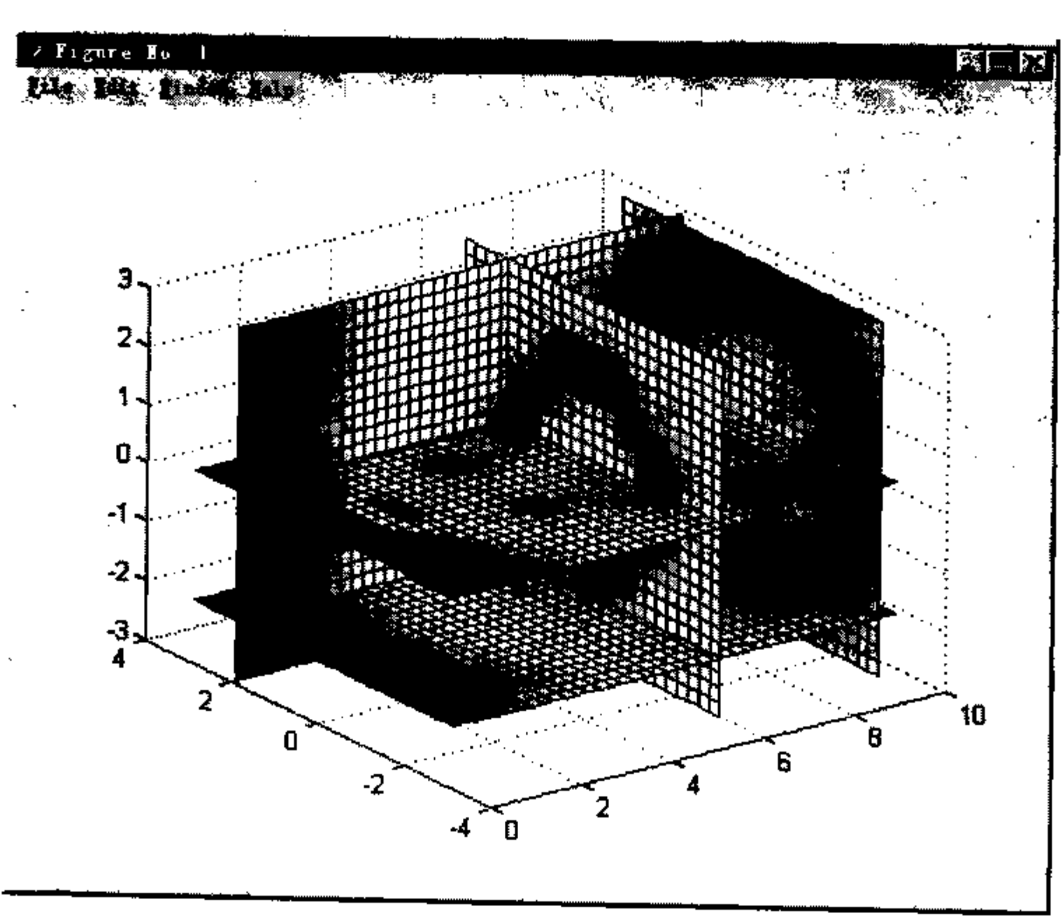


图 8-6　三维插值得到的网格图

5. 一维 FFT 插值

名称：interpft

一维 FFT 插值。

语法：该函数有如下两种表达形式：

- y = interpft(x,n)
- y = interpft(x,n,dim)

描述：前面已经介绍，在 MATLAB 中有两类一维数据的插值方法：多项式插值法和基

于 FFT(快速傅里叶变换)的插值法。

函数 interpft 就是采用一维 FFT 插值。该函数只适合用于周期函数生成数据的插值。它首先计算周期函数的等间距抽样序列形成的向量傅里叶变换，然后加入更多的点计算傅里叶反变换。它的调用方式为：

y = interpft(x,n)

y = interpft(x,n,dim)

其中，x 为抽样序列；n 为需要计算的等间距数；y 为 n 点等间距的计算结果。

举例：

参见函数 interp1。

6. 多维插值

名称：interpn

多维插值。

语法：该函数有如下几种表达形式：

- VI = interpn(X1,X2,X3,...,V,Y1,Y2,Y3,...)
- VI = interpn(V,Y1,Y2,Y3,...)
- VI = interpn(V,ntimes)
- VI = interpn(...,method)

描述：多维插值与三维插值类似。函数 interpn 用来进行多维插值。它的调用格式为：

VI = interpn(X1,X2,X3,...,V,Y1,Y2,Y3,...)

VI = interpn(V,Y1,Y2,Y3,...)

VI = interpn(V,ntimes)

VI = interpn(...,method)

其中 X1、X2、X3…分别是已知 n 维数据点的 x1 坐标矩阵、x2 坐标矩阵、x3 坐标矩阵…；y1、y2、y3…分别为需要插值的点的 y1 坐标矩阵、y2 坐标矩阵、y3 坐标矩阵…；V 为已知点的值；VI 为待插值点的值；参数 method 可以用来选择具体的插值方法，包括：

取 'linear' 时，表示采用线性插值方法(缺省)。

取 'cubic' 时，表示采用立方插值方法。

取 'nearest' 时，表示采用最近点插值方法。

取 'spline' 时，表示采用样条插值方法。

这四种方法都要求把已给数据按 x 做升序或降序排列。

其中，最近点插值法把 n 维空间位置上离插值点最近的已知点的值赋给插值点。线性插值法把已知点用分块线性曲面连接，再对插值点插值。立方插值法将已知点用立方曲面连接后再插值。

举例：

本例将利用 MATLAB 提供的 peaks 函数提取少量数据，比较各种插值方法。

第一步，调用函数[x1,x2]=ndgrid(-3:1:3)生成已知点的 x1 和 x2 坐标：

```
x1 =

    -3    -3    -3    -3    -3    -3    -3
    -2    -2    -2    -2    -2    -2    -2
```

```
     -1    -1    -1    -1    -1    -1    -1
      0     0     0     0     0     0     0
      1     1     1     1     1     1     1
      2     2     2     2     2     2     2
      3     3     3     3     3     3     3
x2 =
     -3    -2    -1     0     1     2     3
     -3    -2    -1     0     1     2     3
     -3    -2    -1     0     1     2     3
     -3    -2    -1     0     1     2     3
     -3    -2    -1     0     1     2     3
     -3    -2    -1     0     1     2     3
     -3    -2    -1     0     1     2     3
```

调用函数 v=peaks(x1,x2)生成已知点的值：

```
v =
    0.0001    0.0007   -0.0088   -0.0365   -0.0137    0.0000    0.0000
    0.0034    0.0468   -0.1301   -1.3327   -0.4808    0.0797    0.0053
   -0.0299   -0.5921    1.8559   -1.6523    0.2289    2.0967    0.1099
   -0.2450   -4.7596   -0.7239    0.9810    3.6886    5.8591    0.2999
   -0.1100   -2.1024   -0.2729    2.9369    2.4338    2.2099    0.1107
   -0.0043   -0.0616    0.4996    1.4122    0.5805    0.1328    0.0057
   -0.0000    0.0004    0.0130    0.0331    0.0125    0.0013    0.0000
```

第二步，调用函数[y1,y2]=ndgrid(-3:0.25:3)，生成需要插值的点的 y1 坐标矩阵和 y2 坐标矩阵。

第三步，调用函数 z1 = interpn(x1, x2, v, y1, y2, 'nearest')，生成采用最近点插值法得到的值，并调用函数 surf(y1, y2, z1)和 contour(y1, y2, z1)，在窗口中显示出得到的表面图和等高线图，如图 8-7 和图 8-8 所示。

第四步，调用函数 z2 = interpn(x1, x2, v, y1, y2, 'linear')，生成采用线性插值法得到的值，并调用函数 surf(y1, y2, z2)和 contour(y1, y2, z2)，在窗口中显示出得到的表面图和等高线图，如图 8-9 和图 8-10 所示。

第五步，调用函数 z3 = interpn(x1, x2, v, y1, y2, 'cubic')，生成采用立方插值法得到的值，并调用函数 surf(y1, y2, z3)和 contour(y1, y2, z3)，在窗口中显示出得到的表面图和等高线图，如图 8-11 和图 8-12 所示。

第六步，调用函数 z4 = interpn(x1, x2, v, y1, y2, 'spline')，生成采用样条插值法得到的值，并调用函数 surf(y1, y2, z4)和 contour(y1, y2, z4)，在窗口中显示出得到的表面图和等高线图，如图 8-13 和图 8-14 所示。

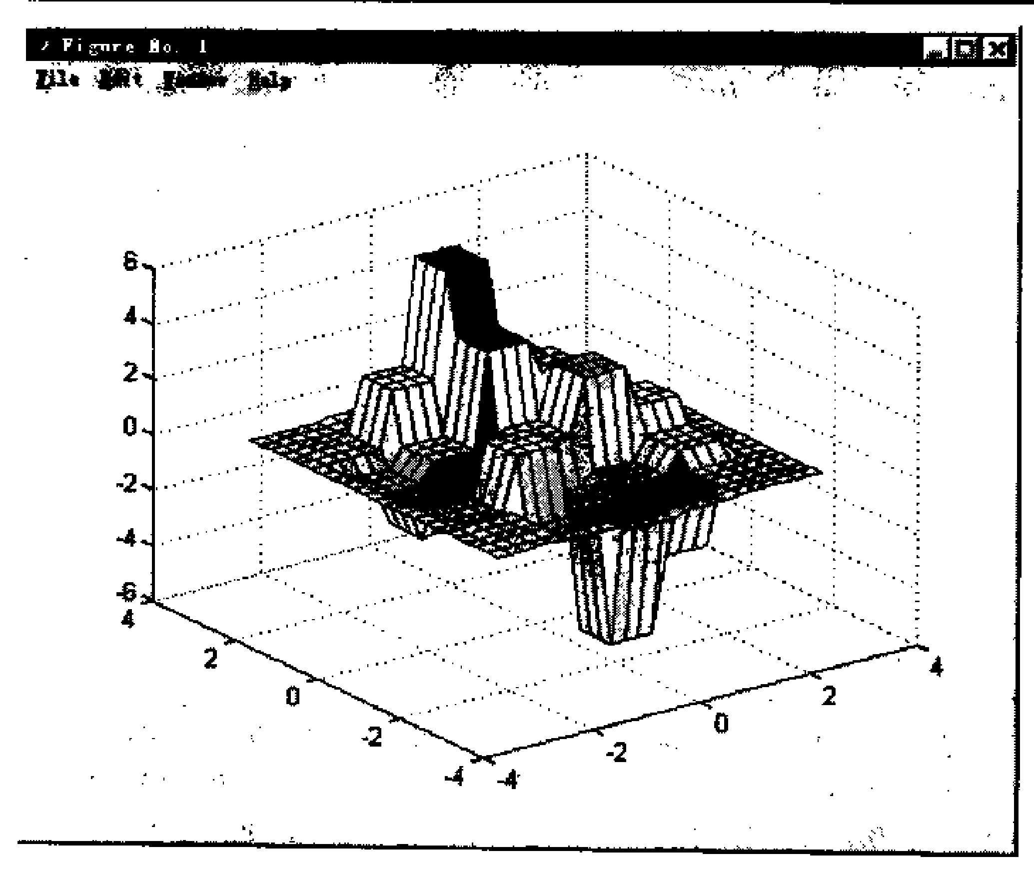


图 8-7 采用最近点插值法得到的表面图

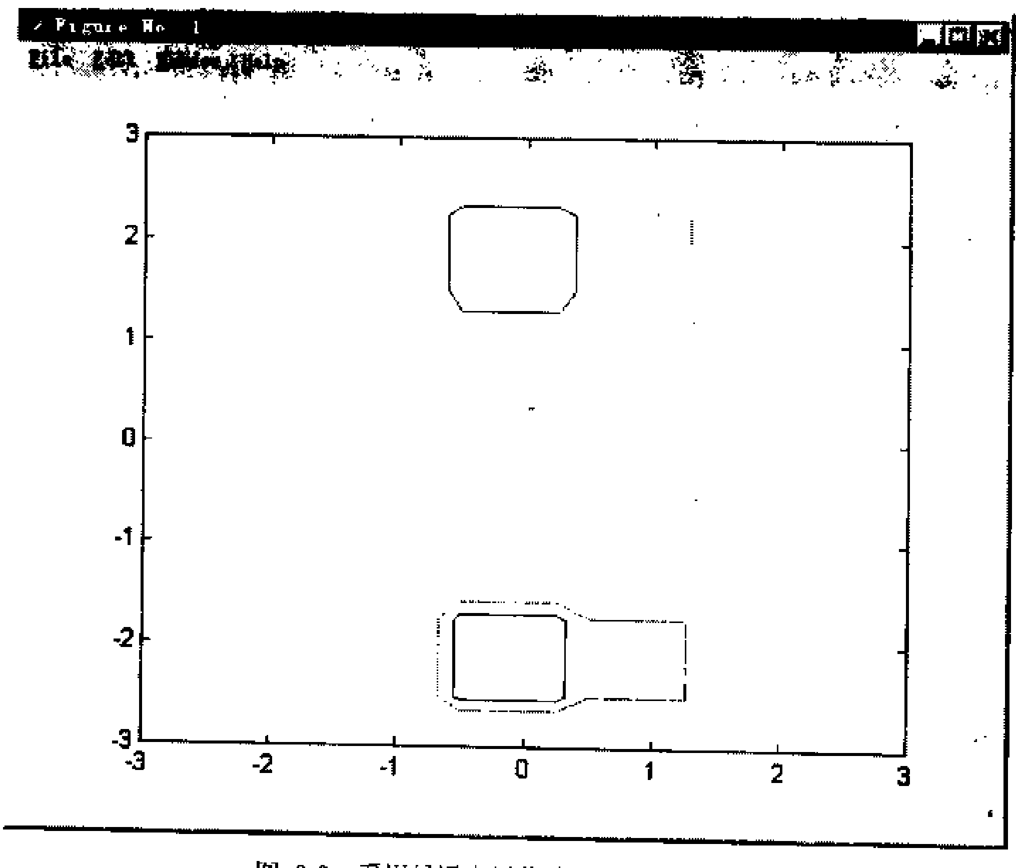


图 8-8 采用最近点插值法得到的等高线图

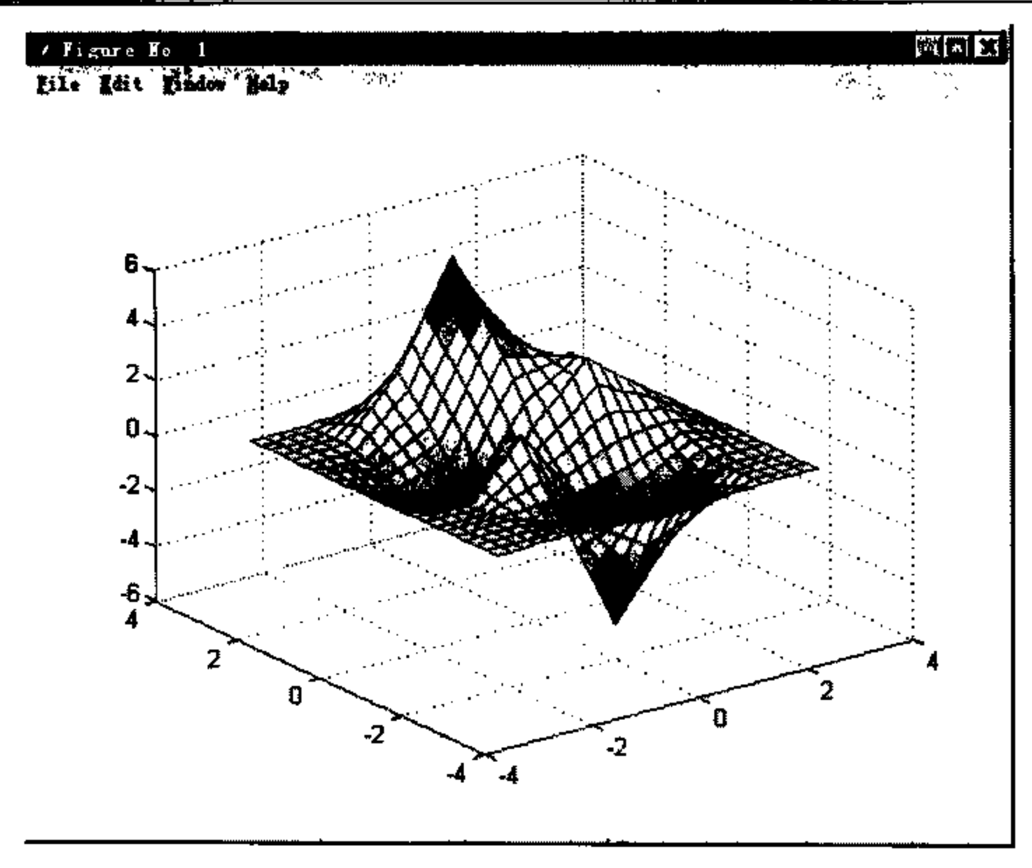


图 8-9　采用线性插值法得到的表面图

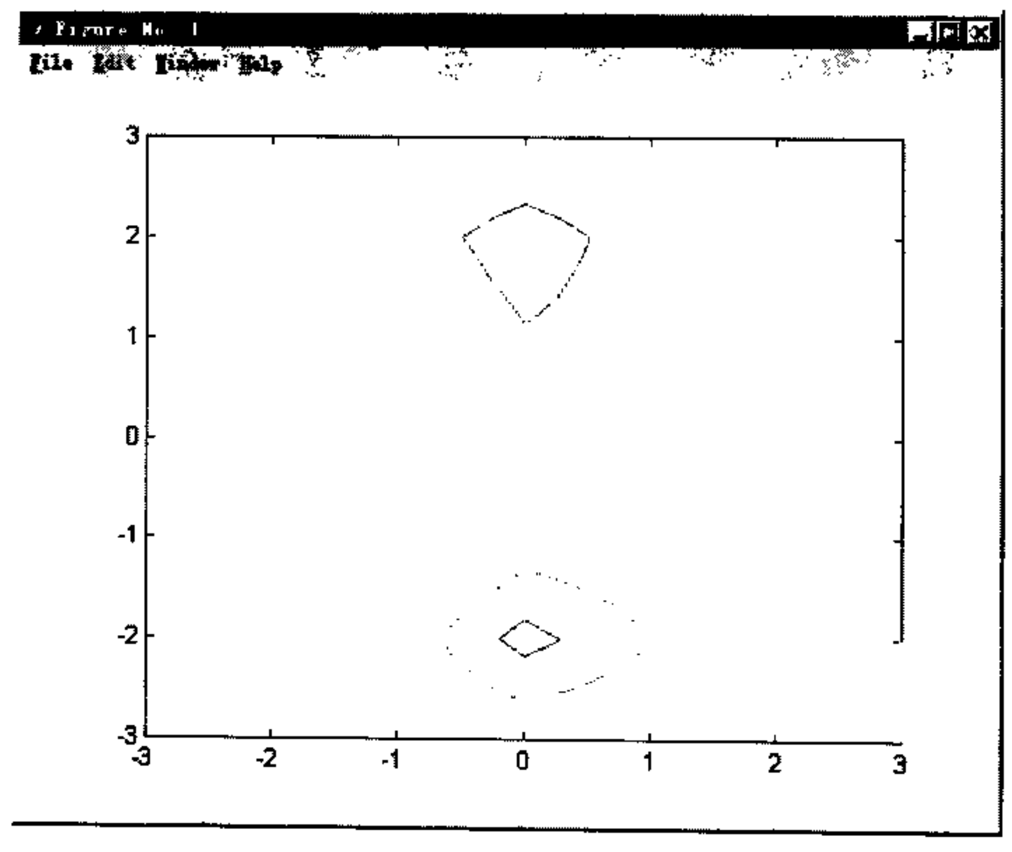


图 8-10　采用线性插值法得到的等高线图

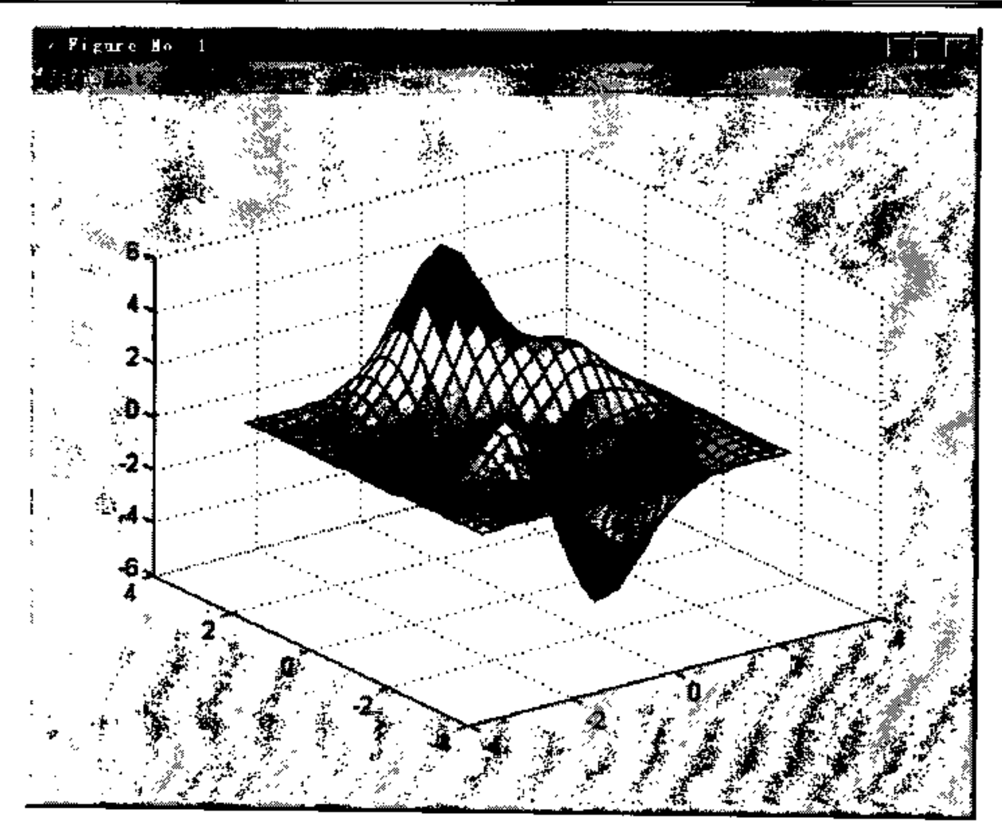


图 8-11　采用立方插值法得到的表面图

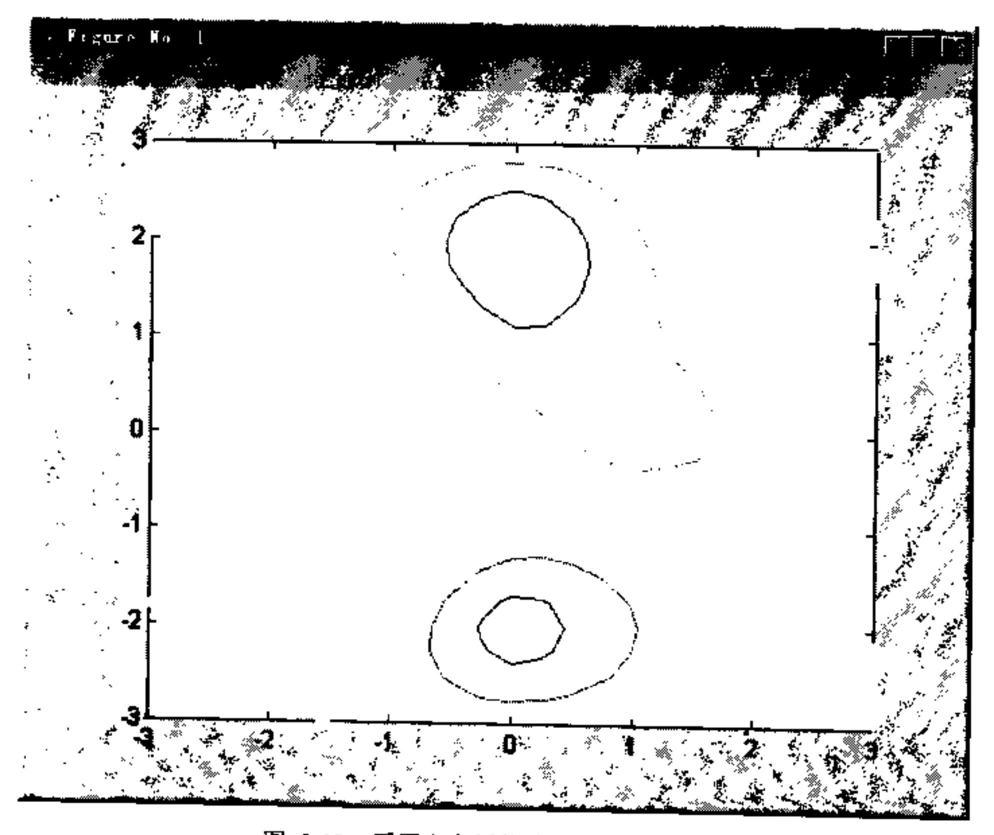


图 8-12　采用立方插值法得到的等高线图

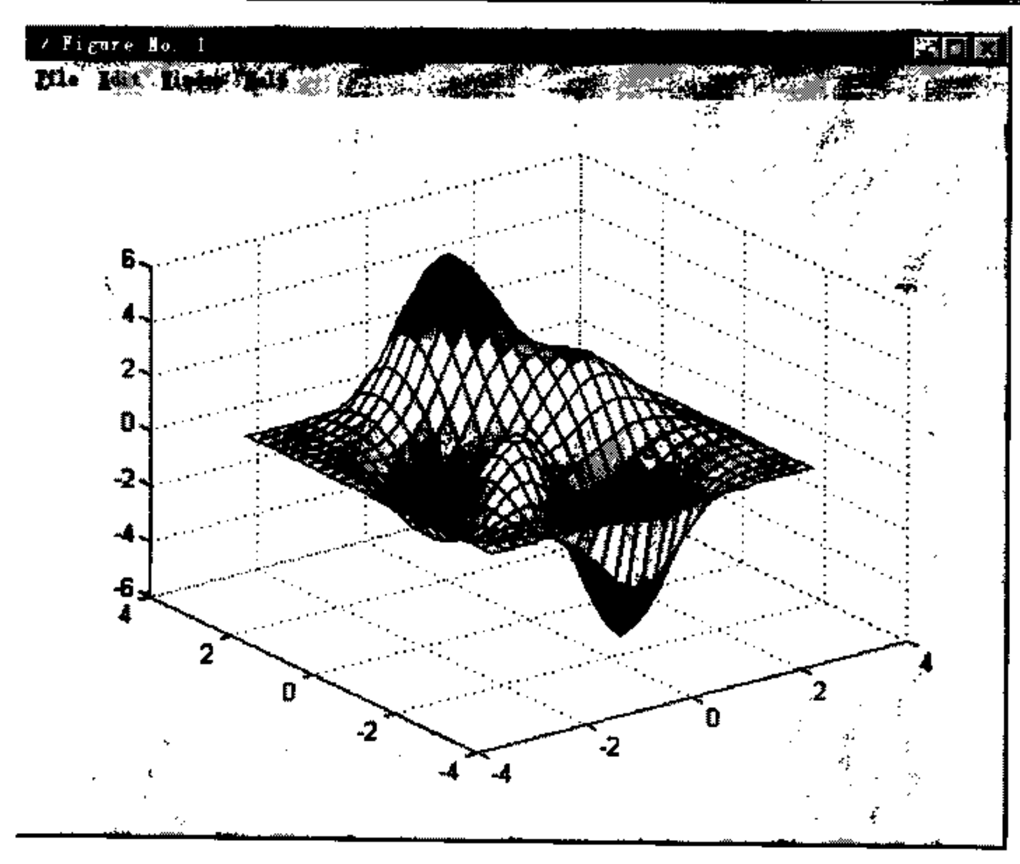


图 8-13　采用样条插值法得到的表面图

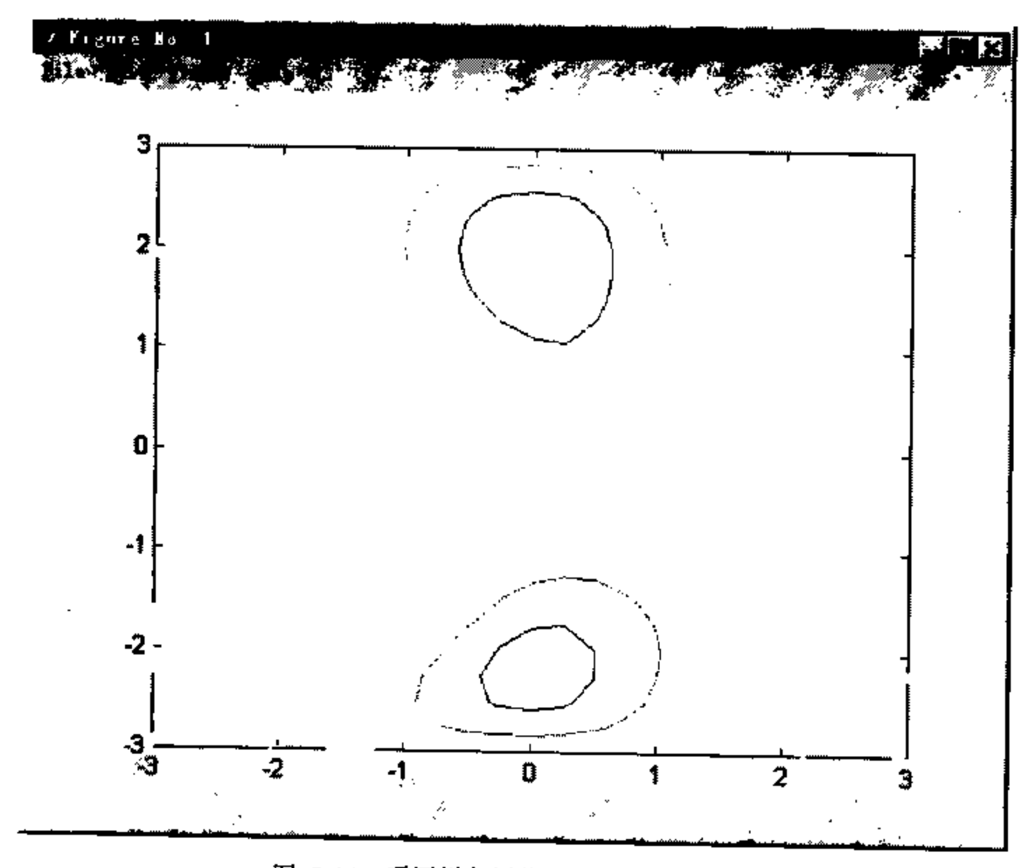


图 8-14　采用样条插值法得到的等高线图

第七步，比较等高线。

采用加密的数据画出图形。调用函数[x1,x2]=ndgrid(-3:0.05:3)和 v=peaks(x1,x2)，生成加密的已知数据点。然后调用函数 surf(x1,x2,v)和 contour(x1,x2,v)将表面图和等高线图在窗口中显示出来。由于网格很密，可认为该表面图和等高线图为真实图形。如图 8-15 和 8-16 所示。

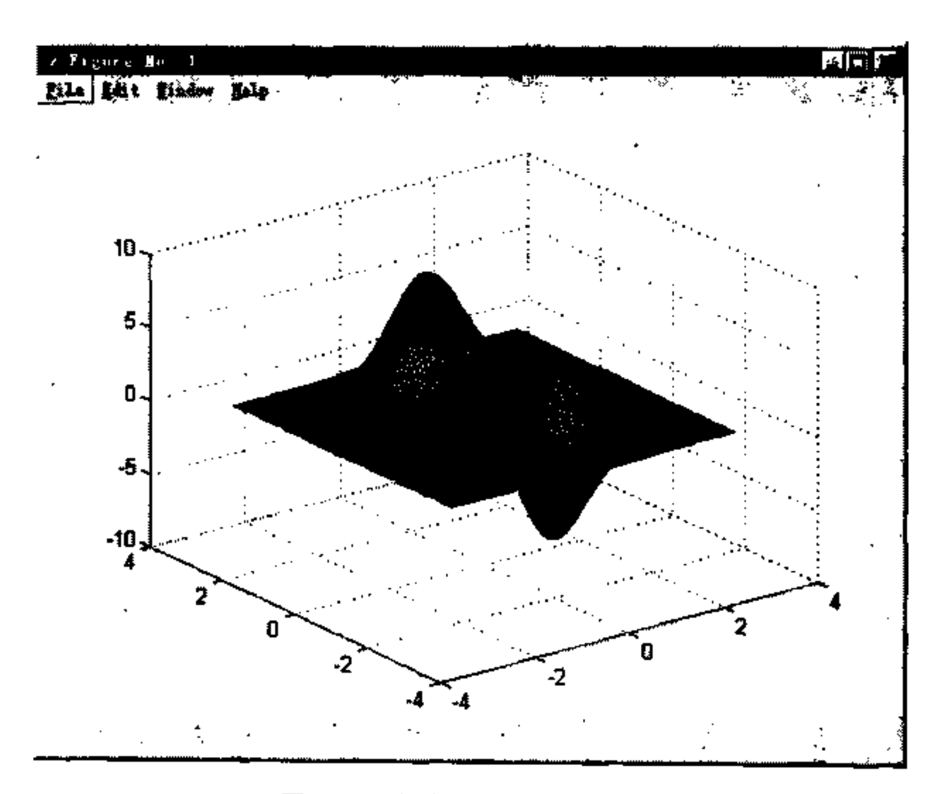


图 8-15 加密网格的真实表面图

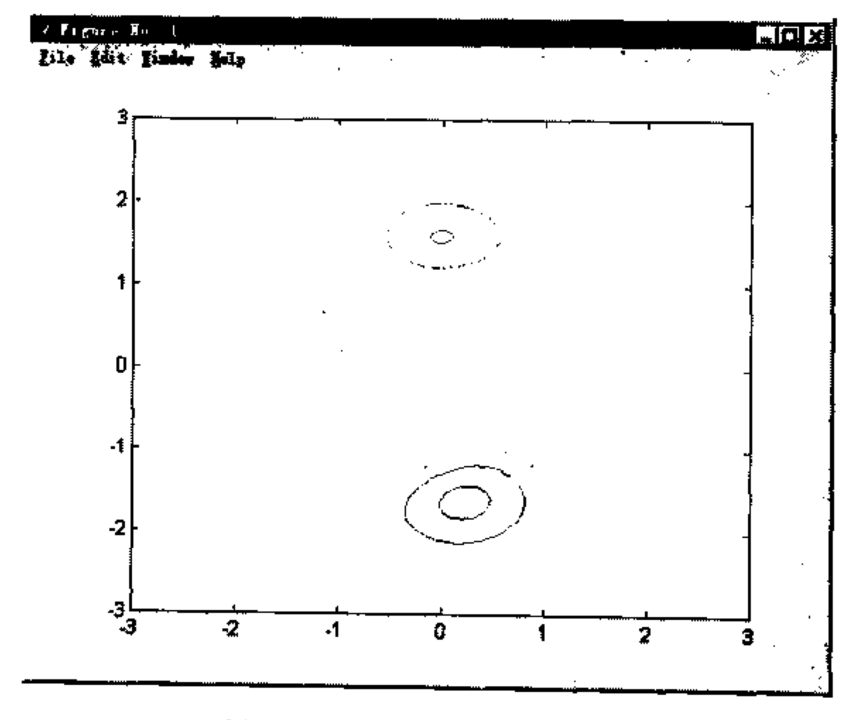


图 8-16 加密网格的真实等高线图

7. 生成三维双数据点的 X 坐标矩阵和 Y 坐标矩阵

名称：meshgrid

生成三维双数据点的 X 坐标矩阵和 Y 坐标矩阵。

语法：该函数有如下几种表达形式：

- [X,Y] = meshgrid(x,y)
- [X,Y] = meshgrid(x)
- [X,Y,Z] = meshgrid(x,y,z)

描述：[X,Y] = meshgrid(x,y)把由向量 x 和 y 所指定的域变换为矩阵 X 和 Y，得到的矩阵可用来计算和绘制三维网格图。X 中的行向量就等于向量 x，Y 中的列向量就等于向量 y。

[X,Y] = meshgrid(x)等价于[X,Y] = meshgrid(x,x)。

[X,Y,Z] = meshgrid(x,y,z)把由向量 x、y 和 z 所指定的域变换为矩阵 X、Y 和 Z，得到的矩阵可用来计算和绘制三维网格图。

举例：

例如调用函数[X,Y] = meshgrid(1:3,10:14)，可以得到：

```
X =
     1     2     3
     1     2     3
     1     2     3
     1     2     3
     1     2     3
Y =
    10    10    10
    11    11    11
    12    12    12
    13    13    13
    14    14    14
```

而调用函数[X,Y] = meshgrid(1:3)可以得到：

```
X =
     1     2     3
     1     2     3
     1     2     3
Y =
     1     1     1
     2     2     2
     3     3     3
```

8. 为多维函数和多维插值准备数据

名称：ndgrid

为多维函数和多维插值准备数据。

语法：该函数有如下两种表达形式：

- [X1,X2,X3,...] = ndgrid(x1,x2,x3,...)
- [X1,X2,...] = ndgrid(x)

描述：[X1,X2,X3,...] = ndgrid(x1,x2,x3,...) 把由向量 x1、x2、x3…所指定的域变换为数组 X1、X2、X3…，得到的数组可用来计算和绘制多维网格图。

[X1,X2,...] = ndgrid(x)等价于[X1,X2,...] = ndgrid(x,x,...)。

举例：

计算函数 $x_1e^{-x_1^2-x_2^2}$ 在区间$-2<x_1<2$ 和$-2<x_2<2$ 内的取值。

首先调用函数[X1,X2] = ndgrid(-2:.2:2, -2:.2:2)，生成所需的数据，然后调用函数 Z = X1 .* exp(-X1.^2 - X2.^2)计算函数的取值。

为了直观起见，调用函数 mesh(Z)，将该函数在窗口中显示出来，如图 8-17 所示。

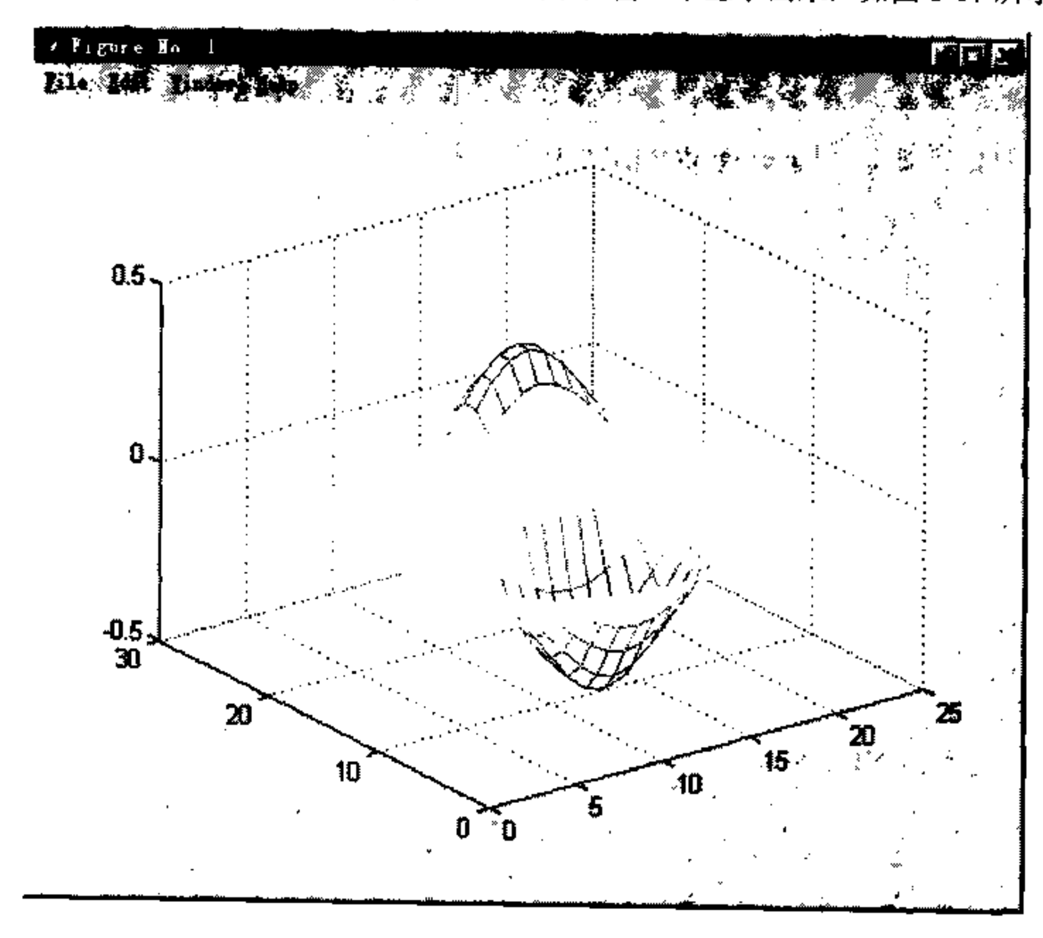


图 8-17 函数 $x_1e^{-x_1^2-x_2^2}$ 的分布图

9. 三次样条插值

名称：spline

三次样条插值。

语法：该函数有如下两种表达形式：

- yy = spline(x,y,xx)
- pp = spline(x,y)

描述：函数 spline 可以用来构造一个样条函数，该样条函数在点 x(j)处的取值为 y(:,j)。

yy = spline(x,y,xx)返回三次样条插值在 xx 处的取值。

pp = spline(x,y)返回三次样条插值的 pp 形式。

举例：

对于如下所示的向量 x 和矩阵 y：

x =0　1.5708　3.1416　4.7124　6.2832

y =

0	1	0	-1	0	1	0
1	0	1	0	-1	0	1

调用函数 pp = spline(x,y)可以得到三次样条插值的 pp 形式，然后执行函数 yy = ppval(pp, linspace(0,2*pi,101))可以得到插值点的取值。

最后调用函数 plot(yy(1,:),yy(2,:),'-b',y(1,2:5),y(2,2:5),'or')和 axis equal 将结果在窗口中显示出来，如图 8-18 所示。

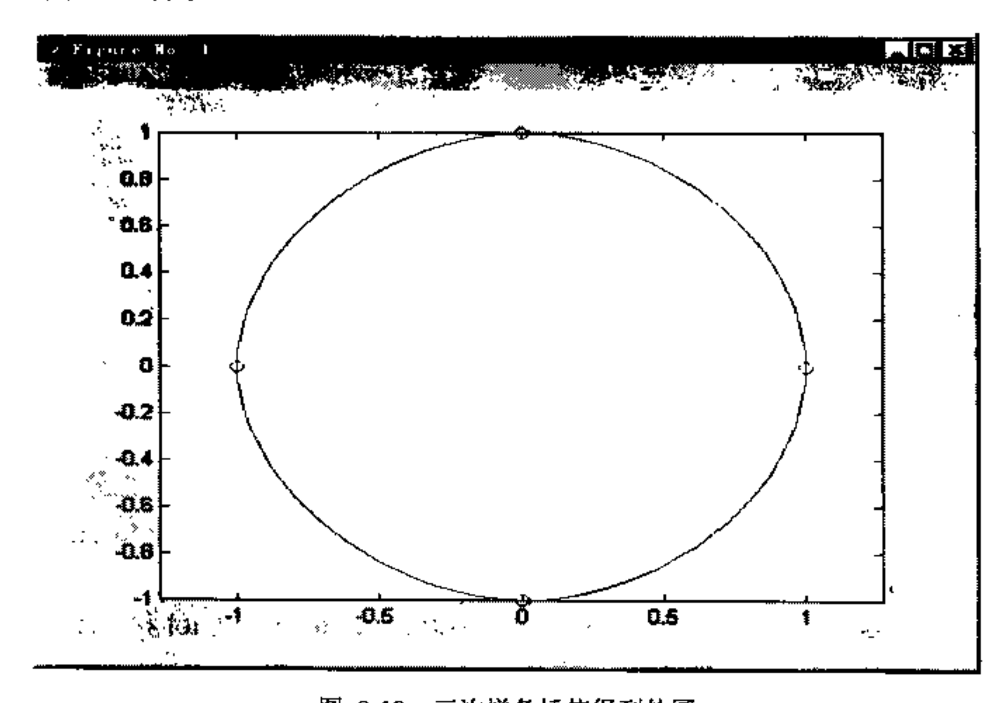


图 8-18　三次样条插值得到的圆

第九章　双重函数与非线性数值方法

9.1　双重函数

双重函数是 MATLAB 特有的一类函数。和其他 MATLAB 函数相比，双重函数的输入参数为数学函数而不是数值矩阵。例如，可用命令 fmin('sin',[-2 2])来求 sin 函数在区间[-2, 2]的最小值。可见 MATLAB 里的双重函数和数学上的复合函数是有区别的。在 MATLAB 中，双重函数主要用于数值积分、非线性方程求解、微分方程求解。

1. 双重数值积分

名称：dblquad

双重数值积分。

语法：该函数有如下几种表达形式：

- result = dblquad('fun',inmin,inmax,outmin,outmax)
- result = dblquad('fun',inmin,inmax,outmin,outmax,tol,trace)
- result = dblquad('fun',inmin,inmax,outmin,outmax,tol,trace,order)

描述：函数 dblquad 用来计算双重数值积分，其调用格式为：

result = dblquad('fun',inmin,inmax,outmin,outmax)利用 quad 法计算双重积分。

result = dblquad('fun',inmin,inmax,outmin,outmax,tol,trace)将 tol 和 trace 传递到 quad 函数，然后计算双重积分。

result = dblquad('fun',inmin,inmax,outmin,outmax,tol,trace,order) 将 tol 和 trace 传递到 quad 函数或 quad8 函数(依赖于字符串 order 的取值)，然后计算双重积分。

其中字符串 fun 为被积函数的函数名，该函数必须是一个包含两个变量、形如 fout=fun(inner,outer)的函数；inmin 和 inmax 为里边一层积分的上下限；outmin 和 outmax 为外边一层的上下限；tol 为误差限；trace 为可选项，当 trace 不为 0 时，将会画出所采用的积分点的图形；order 用来指定积分方法(默认方法为 quad 法)，它的取值可以为 'quad' 或 'quad8'，也可以采用用户自己编写的积分方法。

举例：

用 M 文件 integrnd.m 生成被积函数：

```
function out = integrnd(x, y)
out = y*sin(x)+x*cos(y);
```

调用函数 result = dblquad('integrnd',pi,2*pi,0,pi)计算双重积分，可得 result = -9.8698。

调用函数 result = dblquad('integrnd',pi,2*pi,0,pi,[],'quad8')，采用 quad8 法计算双重积分，可得：result = -9.8696。

2. 单变量函数的最小值

名称：fminbnd

求单变量函数的最小值。

语法：该函数有如下几种表达形式：

- x = fminbnd(fun,x1,x2)
- x = fminbnd(fun,x1,x2,options)
- x = fminbnd(fun,x1,x2,options,P1,P2,...)
- [x,fval] = fminbnd(...)
- [x,fval,exitflag] = fminbnd(...)
- [x,fval,exitflag,output] = fminbnd(...)

描述：函数 fminbnd 用于求指定区间上单变量函数的局部极小值。其调用格式为：

x = fminbnd(fun,x1,x2)返回区间 x1 < x < x2 内，函数 fun 的极小值。

x = fminbnd(fun,x1,x2,options)利用结构 options 提供的参数求函数 fun 的极小值。

x = fminbnd(fun,x1,x2,options,P1,P2,...)将附加参数 P1, P2…传递给目标函数。

[x.fval] = fminbnd(...)返回目标函数在 x 处的取值。

[x.fval,exitflag] = fminbnd(...)返回描述 fminbnd 函数退出条件的 exitflag 值。

[x,fval,exitflag,output] = fminbnd(...)返回一个包含优化信息的结构 output。

其中字符串 fun 为函数名；x1 和 x2 指定求极小值的区间；options 为可选项，用来设置是否显示中间结果、设置以自变量或函数值迭代误差控制的迭代条件、最大迭代步数，用户可以使用 optimset 函数来定义 options 结构中的参数：

Display 参数用来指定显示的级别。取 off 时，表示不输出；取 iter 时，表示在每一迭代步都要输出显示；取 final 时，表示仅仅显示最后的结果。

MaxFunEvals 参数用来指定函数计算允许的最大数目。

MaxIter 参数用来指定最大允许迭代次数。

TolX 参数用来给定收敛允许值。

如果没有选项设置，则令 options=[]。

参数 exitflag 用来描述函数 fminbnd 的退出条件：

当 exitflag>0 时，表示函数收敛于 x。

当 exitflag=0 时，表示已经达到了函数计算允许的最大数目。

结构 output 中包含一些有关优化的信息：

output.algorithm 表示所使用的优化算法。

output.funcCount 表示函数计算的数目。

output.iterations 表示已经迭代的次数。

举例：

本例求解函数 $f(x)=x^3-2x-5$ 在区间(0, 2)上的极小值。

首先利用函数 f = inline('x.^3-2*x-5')建立内置目标函数。然后调用函数 x = fminbnd(f, 0, 2)求解其极小值，可得：x = 0.8165。而且函数在最小值处的取值为 y = f(x)= -6.0887。

调用函数 x = fminbnd(f, 0, 2,1)可以显示函数的运行步骤：

```
Func evals      x              f(x)          Procedure
```

1	0.763932	-6.08204	initial
2	1.23607	-5.58359	golden
3	0.472136	-5.83903	golden
4	0.786475	-6.08648	parabolic
5	0.823917	-6.08853	parabolic
6	0.8167	-6.08866	parabolic
7	0.81645	-6.08866	parabolic
8	0.816497	-6.08866	parabolic
9	0.81653	-6.08866	parabolic

x = 0.8165。

可见，本例共迭代了 8 次，其中，Procedure 项为迭代使用的方法。

9.2 非线性数值方法

1. 多变量函数的最小值

名称：fminsearch

求多变量函数的最小值。

语法：该函数有如下几种表达形式：

- x = fminsearch(fun,x0)
- x = fminsearch(fun,x0,options)
- x = fminsearch(fun,x0,options,P1,P2,...)
- [x,fval] = fminsearch(...)
- [x,fval,exitflag] = fminsearch(...)
- [x,fval,exitflag,output] = fminsearch(...)

描述：和函数 fminbnd 类似，函数 fminsearch 用于求多变量函数的极小值。二者的不同之处在于，调用 fminbnd 时给定求极值的区间，而调用 fminsearch 时，应指定初始极值向量。fminsearch 函数将计算出所给向量附近的一个极小值。其调用格式为：

x = fminsearch(fun,x0)返回初始向量 x0 附近、使函数 fun 取局部极小值的向量 x。

x = fminsearch(fun,x0,options)利用结构 options 提供的优化参数求函数 fun 的极小值。

x = fminsearch(fun,x0,options,P1,P2,...) 将附加参数 P1, P2...传递给目标函数。

[x,fval] = fminsearch(...)返回目标函数在 x 处的取值。

[x,fval,exitflag] = fminsearch(...)返回描述 fminsearch 函数退出条件的 exitflag 值。

[x,fval,exitflag,output] = fminsearch(...)返回一个包含优化信息的结构 output。

其中字符串 fun 为函数名；x0 为给定的初始向量；options 为可选项，用来设置是否显示中间结果、设置以自变量或函数值迭代误差控制的迭代条件、最大迭代步数，用户可以使用 optimset 函数来定义 options 结构中的参数：

Display 参数用来指定显示的级别。取 off 时，表示不输出；取 iter 时，表示在每一迭代步都要输出显示；取 final 时，表示仅仅显示最后的结果。

MaxFunEvals 参数用来指定函数计算允许的最大数目。

MaxIter 参数用来指定最大允许迭代次数。

TolFun 参数用来给定的函数的收敛允许值。

TolX 参数用来给定的 x 的收敛允许值。

如果没有选项设置，则令 options=[]。

参数 exitflag 用来描述函数 fminsearch 的退出条件：

当 exitflag>0 时，表示函数收敛于 x。

当 exitflag=0 时，表示已经达到了函数计算允许的最大数目。

当 exitflag<0 时，表示函数没有收敛于 x。

结构 output 中包含一些有关优化的信息：

output.algorithm 表示所使用的优化算法。

output.funcCount 表示函数计算的数目。

output.iterations 表示已经迭代的次数。

举例：

求解函数 $f(x)=100(x_2-x_1^2)^2+(1-x_1)^2$ 的极小值。

首先在 M 文件 banana.m 中定义函数：

```
function f = banana(x)
f = 100*(x(2)-x(1)^2)^2+(1-x(1))^2;
```

调用函数[x,fval] = fminsearch('banana',[-1.2, 1])求解函数的极小值，得到：

x =1.0000　　1.0000，

fval = 8.1777e-010。

2. 单变量函数求 0 值

名称：fzero

单变量函数求 0 值。

语法：该函数有如下几种表达形式：

- x = fzero(fun,x0)
- x = fzero(fun,x0,options)
- x = fzero(fun,x0,options,P1,P2,...)
- [x,fval] = fzero(...)
- [x,fval,exitflag] = fzero(...)
- [x,fval,exitflag,output] = fzero(...)

描述：函数 fzero 用于找单变量函数值为 0 时的自变量的值。调用函数 fzero 时，需要指定初始值。函数会自动在初始值附近寻找一个与端点函数异号的区间，然后通过迭代求函数值为 0 时对应的自变量；当找不到这样的区间时，返回 NaN。当已知端点函数值异号的区间，输入该区间，一定可以求出一个 0 值。其调用格式为：

x = fzero(fun,x0)返回在 x0 附近使函数 fun 取 0 值的 x。

x = fzero(fun,x0,options)利用结构 options 提供的优化参数进行求解。

x = fzero(fun,x0,options,P1,P2,...)将附加参数 P1, P2…传递给目标函数。

[x,fval] = fzero(...)返回目标函数在 x 处的取值。

[x,fval,exitflag] = fzero(...)返回描述 fzero 函数退出条件的 exitflag 值。

[x,fval,exitflag,output] = fzero(...)返回一个包含优化信息的结构 output。

其中字符串 fun 为函数名；x0 为给定的初始值；options 为可选项，用来设置是否显示中间结果、设置以自变量或函数值迭代误差控制的迭代条件，用户可以使用 optimset 函数来定义 options 结构中的参数：

Display 参数用来指定显示的级别。取 off 表示不输出；取 iter 表示在每一迭代步都要输出显示；取 final 表示仅仅显示最后的结果。

TolX 参数用来给定的 x 的收敛允许值。

如果没有选项设置，则令 options=[]。

参数 exitflag 用来描述函数 fzero 的退出条件：

当 exitflag>0 时，表示函数找到取 0 值的 x。

当 exitflag<0 时，表示函数没有找到需要的 x。

结构 output 中包含一些有关优化的信息：

output.algorithm 表示所使用的优化算法。

output.funcCount 表示函数计算的数目。

output.iterations 表示已经迭代的次数。

需要指出的是，对于利用函数 fminbnd、fminserarch 和 fzero 求函数的极值点和 0 值点的优化问题，经常要通过多次迭代才能得到结果。因此，当初始值或区间取得好时，函数的运行效率可以大大提高。解复杂问题时，先选用同类型的低阶问题(变量较少)求解，然后找出问题的极值点，作为高阶问题的初始值，这种思路往往能够避开局部极值，而求出全局最优解。下面为一些常见问题的解决办法。

如前所述，函数 fminbnd 和 fminserarch 求出的值为局部最小值，而非全局最小值。要求全局最小值，应该多选择几个初始值和区间。对于有些优化问题，当取得最优值时，函数无意义，这时可用复函数来修改函数。

当利用函数 fminbnd、fminserarch 和 fzero 求解最优化问题时，有时会陷入死循环，或求得的值不是极小值(用 fzero 时不为 0)，这时应该检查求解过程中，是否有复数或其他的数据形式出现。因为这三个函数只是求解实数情况，对于其他数据可能得到莫名其妙的结果。在数据检查时，函数 isreal 和 isfinite 特别有用。

举例：

本例寻找使函数 $f(x)=-x^3-2x-5$ 取 0 时相对应的自变量 x。

首先在 M 文件 f.m 中定义函数：

```
function y = f(x)
y = x.^3-2*x-5;
```

然后调用函数 x = fzero('f',2)，寻找 2 附近使函数 f 取极小值的 x，可得：x =2.0946。

为了显示迭代过程，可以调用函数 x = fzero('f',2,[],1)，得到：

```
Func evals        x           f(x)        Procedure
     1            2            -1          initial
```

```
    2          1.94343       -1.54667         search
    3          2.05657       -0.414934        search
    4             1.92       -1.76211         search
    5             2.08       -0.161088        search
    6          1.88686       -2.05602         search
    7          2.11314       0.209619         search
 Looking for a zero in the interval [1.8869, 2.1131]
    8          2.0922        -0.0261891       interpolation
    9          2.09453   -0.000272594         interpolation
   10          2.09455   6.41518e-009         interpolation
   11          2.09455   -8.9706e-014         interpolation
   12          2.09455 -8.88178e-016          interpolation
   13          2.09455   9.76996e-015         interpolation
```

x = 2.0946。

从上可以看出，在前面的 6 个迭代步中，函数 fzero 首先寻找合适的区间(端点函数值异号的区间)。确定了区间之后，开始进一步迭代求解，直到求出使函数值取 0 值的 x 为止。

在上例中，函数的输入参数中有一个空矩阵，表示采用默认的误差限。在迭代方法上，既有双分法，也有插值法。

3. 解微分方程

名称：ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb

解微分方程。

语法：该函数有如下几种表达形式：

- [T,Y] = solver('F',tspan,y0)
- [T,Y] = solver('F',tspan,y0,options)
- [T,Y] = solver('F',tspan,y0,options,p1,p2...)
- [T,Y,TE,YE,IE] = solver('F',tspan,y0,options)

描述：

在工程中，常微分方程的初值问题是比较常见的一类问题。常微分方程的初值问题的标准数学表述为：

$$\begin{cases} y' = F(t,y), a \le t \le b \\ y(a) = y_0 \end{cases}$$，其中 y 为向量。

任何高阶的常微分方程都可以用替换法化为上式所示的一阶形式。

在 MATLAB 里调用函数求解常微分方程的步骤为：

1，化方程组为标准形式。例如

$y''' - 3y'' - y'y = 0$， $y(0) = 0$， $y'(0) = 1$， $y''(0) = -1$。

把微分方程的高阶导数写为低阶导数的算式，即 $y''' = 3y'' + y'y$。令 $y_1 = y$， $y_2 = y'$， $y_3 = y'''$，则原方程化为下列方程组：

$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ y_3' = 3y_3 + y_2y_1 \end{cases}$，满足初始条件$\begin{cases} y_1(0) = 0 \\ y_2(0) = 1 \\ y_3(0) = -1 \end{cases}$，已把该方程化成了标准形式。

2，把微分方程编写成 M 文件。例如上面方程对应的 M 文件内容如下：

```
function dy=F(t,y)
dy=[y(2);y(3);3*y(3)+y(2)*y(1) ]
```

在 M 文件中，虽然写微分方程时并不同时包含参数 t 和 y，但第一行必须包含这两个输入变量，向量 dy 必须是列向量。

3，调用一个微分方程的求解函数进行求解。其格式为：

[T,Y] = solver('F',tspan,y0)对于初始条件为 y0，积分范围为 tspan = [t0 tfinal]的微分方程 y' = F(t,y)进行积分。

[T,Y] = solver('F',tspan,y0,options) 对于初始条件为 y0，积分范围为 tspan = [t0 tfinal]的微分方程 y' = F(t,y)进行积分。其中一些属性的取值要用 options 结构中给定的值进行替换。常用的一些属性包括：相对误差允许值 RelTol（标量），缺省值为 1e-6；绝对误差允许值 AbsTol(向量)，缺省时，每个元素的取值均为 1e-6。

[T,Y] = solver('F',tspan,y0,options,p1,p2...)对于初始条件为 y0，积分范围为 tspan = [t0 tfinal]的微分方程 y' = F(t,y)进行积分。其中一些属性的取值要用 options 结构中给定的值进行替换。并将附加参数 p1,p2...传递到 M 文件。

[T,Y,TE,YE,IE] = solver('F',tspan,y0,options)当 options 结构中的属性 Events 被设置为 on 时，该函数对初始条件为 y0，积分范围为 tspan = [t0 tfinal]的微分方程 y' = F(t,y)进行积分，同时寻找 ODE 文件中定义的一个事件函数的零交叉点。并将其中一些属性的取值用 options 结构中给定的值进行替换。用户必须对 ODE 文件进行编辑，从而使得 F(t,y,'events')能够返回相应的信息。

其中 F 为包含微分方程的 M 文件名；tspan 为积分的数据范围，其格式为[t0 tfinal]。积分时，从 t0 积到 tfinal。如果想得到特定时间点的解，可以使用 tspan = [t0,t1, ..., tfinal]；y0 为 t0 时刻的初始条件向量；options 为由函数 odeset 创建的可选积分参数，具体内容可以参见 odeset 中有关说明；p1,p2...为传递给 F 的可选参数。

输出参数 T 为时刻列向量；Y 表示不同时刻的函数值组成的矩阵；TE 是一个描述事件发生时间的列向量；YE 为矩阵，其中的行向量为相应于 TE 中元素的解；IE 是一个包含已发生事件索引的向量。

若用户没有指定输出参数，求解器会调用缺省的输出函数 odeplot 将计算结果在窗口中绘制出来。也可以通过将 OutputFcn 属性的值设置为 odeplot 来实现这一功能。另外将 OutputFcn 属性的取值设置为 odephas2 或 odephas3，可分别用来绘制二维或三维相位平面。详细情况可参见 odefile 中的说明。

MATLAB 利用数值方法来求解常微分方程的解，其求解的大致思路为：把求解的时间区间划分成有限步，对应于每一步将计算出一个解，如果求得的解不满足误差限制，则减小步长，再求解，如此过程循环往复，直到满足误差限为止。

当常微分方程化为标准型以后，一般来说都是方程组。有些方程组的解的不同分量的数量级差别较大，这对于数值求解是一大困难。这种问题被称为刚性(stiff)问题。在化学反

应、电子网格和自动控制等领域都是常见的。MATLAB 即能够解决非刚性问题，又能够解决刚性问题。

MATLAB 提供了三个解决非刚性问题的函数：ode45、ode23 和 ode113。其中：

1，函数 ode45 基于显式的龙格—库塔(4,5)法，采用单步法来计算。在计算时，只需要前一个点的计算结果。由于该函数计算较快，所以可以作为解题时的“初试”法。

2，函数 ode23 是基于显式的龙格—库塔(2,3)的单步法。适用于误差限较大、稍带刚性的问题。ode23 比函数 ode45 更有效。

3，函数 ode113 采用 Adams-Bashforth-Moulton 方法，当误差限要求严格时，该函数比 ode45 函数更有效。ode113 采用多步法，计算时，同时需要前几个点的计算结果。

由于刚性方程的特殊性，用解非刚性方程的函数求解刚性方程往往会得出完全错误的答案。MATLAB 提供了四个解刚性方程的函数：ode15s、ode23s、ode23t 和 ode23tb。其中：

1，ode15s 采用数值差分方法，通常采用的 Gear 法为多步法，而且计算精度较低。

2，ode23s 采用二阶改进的 Rosenbrock 方法。由于 ode23s 采用单步法，对于比较宽松的误差限，ode23s 往往能够得出比 ode15s 更快、更好的结果。

3，ode23t 基于“自由”插值的梯形规则。当所求解的微分方程系统具有适度刚性并且用户希望得到一个不包含数值阻尼的解时，可以使用该求解器。

4，ode23tb 基于隐式的龙格—库塔公式 TR-BDF2。其中第一步采用梯形规则进行求解；第二步采用一个二阶后差分公式进行求解。同时经过适当的构造和修正，使得在这两步中使用完全相同的迭代矩阵。与 ode23s 求解器类似，当误差限比较宽松时，采用 ode23tb 求解器比采用 ode15s 求解器具有更高的计算效率。

所有的求解器都可以求解形如 $y'=F(t,y)$ 的微分方程。其中 ode15s、ode23s、ode23t 和 ode23tb 可以求解形如 $My'=F(t,y)$ 的微分方程，而且 ode15s、ode23t 和 ode23tb 还可以求解形如 $M(t)y'=F(t,y)$ 的微分方程。表 9-1 中列出了各个求解器的适用范围和计算精度。

表 9-1　求解器的适用范围和计算精度

求解器	求解问题的类型	计算的精度	使用条件
ode45	非刚性	中	大多数情况下均使用该求解器。
ode23	非刚性	低	适用于比较宽松的误差限，或求解适度刚性问题。
ode113	非刚性	低—高	适用于对误差限要求比较严厉的情况，或者求解一个计算量比较大的 ODE 文件。
ode15s	刚性	低—中	当采用 ode45 计算速度太慢，或包含质量矩阵。
ode23s	刚性	低	如果采用比较宽松的误差限来求解刚性系统，或包含一个常质量矩阵。
ode23t	适度刚性	低	如果所需要求解的是一个适度刚性问题，而且得到一个不含数值阻尼的解。
ode23tb	刚性	低	如果采用比较宽松的误差限来求解刚性系统，或包含一个质量矩阵。

表 9-2　　求解器接收的参数

参数	Ode45	ode23	ode113	Ode15s	ode23s	ode23t	ode23tb
RelTol, AbsTol	√	√	√	√	√	√	√
OutputFcn,OutputSel, Refine, Stats	√	√	√	√	√	√	√
Events	√	√	√	√	√	√	√
MaxStep, InitialStep	√	√	√	√	√	√	√
Jconstant, Jacobian, Jpattern, Vectorized				√	√	√	√
Mass, MassSingular	√	√	√	√	√	√	√
MaxOrder, BDF				√		√	√

需要指出的是，不同的求解器接收选项结构中的不同参数。表 9-2 中列出了各个求解器中所接收的参数的情况。

举例：

例 1 是一个描述不受外力作用的刚体的运动方程，该方程系统是一个非刚性系统。如下所示：

$$y_1' = y_2 y_3, \quad y_1(0) = 0$$

$$y_2' = -y_1 y_3, \quad y_2(0) = 1$$

$$y_3' = -0.51 y_1 y_2, \quad y_3(0) = 1$$

为了模拟该系统，首先在 M 文件 rigid.m 中创建如下所示的函数：

```
function dy = rigid(t,y)
dy = zeros(3,1);      % a column vector
dy(1) = y(2) * y(3);
dy(2) = -y(1) * y(3);
dy(3) = -0.51 * y(1) * y(2);
```

利用函数 options = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-5]) 设置选项结构中的有关参数。结果为：

```
options =
          AbsTol: [1.0000e-004 1.0000e-004 1.0000e-005]
             BDF: []
          Events: []
     InitialStep: []
        Jacobian: []
       JConstant: []
        JPattern: []
            Mass: []
    MassConstant: []
        MaxOrder: []
         MaxStep: []
     NormControl: []
       OutputFcn: []
       OutputSel: []
          Refine: []
          RelTol: 1.0000e-004
           Stats: []
```

Vectorized: []

然后调用函数[T,Y] = ode45('rigid',[0 12],[0 1 1],options)对微分方程进行求解。

为了直观起见，调用函数 plot(T,Y(:,1),'-',T,Y(:,2),'-.',T,Y(:,3),'.')将求得的结果在窗口中显示出来，如图 9-1 所示。

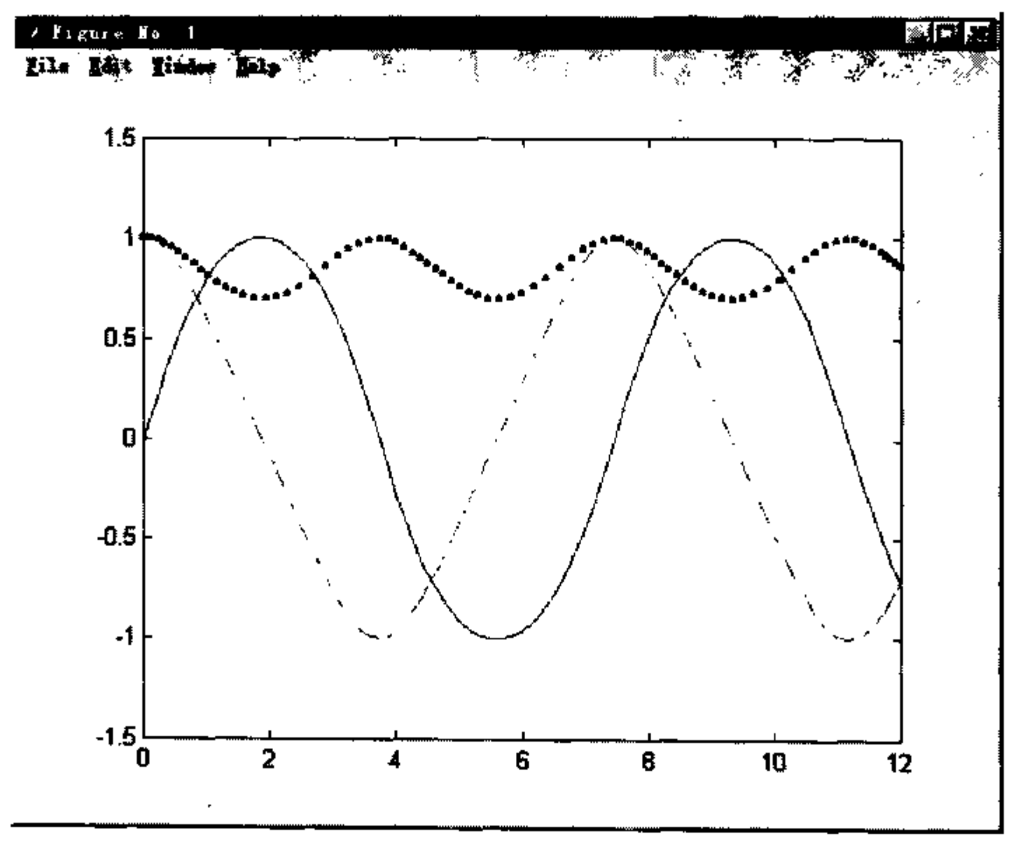


图 9-1　微分方程解的曲线

例 2 是由描述松弛振动过程的 van der Pol 方程得到的一个刚性系统。其表达式如下所示：

$y_1' = y_2$，$y_1(0) = 0$

$y_2' = 1000(1 - y_1^2)y_2 - y_1$，$y_2(0) = 1$。

为了模拟该系统，首先在 M 文件 vdp1000.m 中创建如下所示的函数：

```
function dy = vdp1000(t,y)
dy = zeros(2,1);      % a column vector
dy(1) = y(2);
dy(2) = 1000*(1 - y(1)^2)*y(2) - y(1);
```

在该例中，我们将使用缺省的相对误差限 RelTol(1e-3)和绝对误差限 AbsTol(1e-6)，在时间区间[0 3000]内进行求解，初始向量为[2 0]。因此调用函数[T,Y] = ode15s('vdp1000',[0 3000],[2 0])。

最后调用函数 plot(T,Y(:,1),'-o')将结果在窗口中显示出来，如图 9-2 所示。

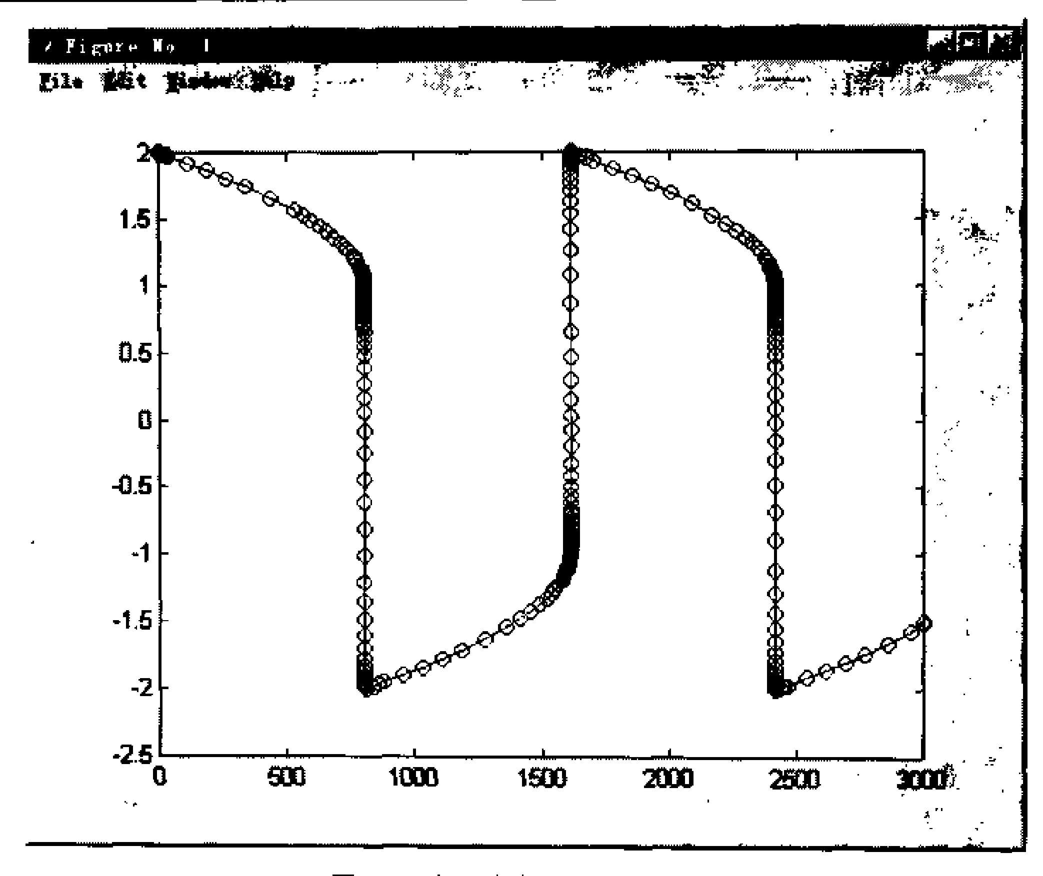


图 9-2 例 2 中微分方程解的曲线

4. 为 ODE 求解器定义微分方程

名称：odefile

为 ODE 求解器定义一个微分方程。

描述：odefile 并不是一个命令或函数。它只是一个描述如何创建 M 文件的帮助条目。该 M 文件用来定义所要求解的方程系统。事实上，定义方程系统是使用 MATLAB 提供的任一求解器进行求解的第一步。

用户可以使用 odefile M 文件来定义一个如下所示的微分方程系统：

$Y' = F(t,Y)$

或 $M(t,Y)Y' = F(t,Y)$。

其中，t 为自变量(标量)，通常是指时间。Y 是一个因变量(向量)。F 是 t 和 Y 的函数，而且是一个和向量 Y 具有相同长度的列向量。M(t,Y)是一个依赖于时间和状态的质量矩阵。

在 MATLAB 所提供的 ODE 求解器中，除了 ode23s 之外，都可以求解形如 $M(t,Y)Y' = F(t,Y)$ 的微分方程。ode23s 求解器则只能求解质量矩阵为常数的情形。其中，ode15s 和 ode23t 求解器还可以求解一些形如 $M(t)Y' = F(t,Y)$ 的微分代数方程(DAEs)。

除了可以定义微分方程系统外，用户还可以在 ODE M 文件中给定一个完整的初值问题，从而就可以不必在命令行中输入时间和初始值。

下面介绍如何使用 ODE 文件模板：

首先在 MATLAB 窗口中输入 help odefile 命令，此时会出现如下所示的帮助条目：

```
ODEFILE ODE file syntax.
  An ODE file is an M-file function you write to define a differential
```

equation problem for the ODE Suite solvers. This M-file is referred to as 'ODEFILE' here, although you can give your M-file any name you like.

By default, the ODE Suite solvers solve initial value problems of the form dy/dt = F(t,y) where t is an independent variable and y is a vector of dependent variables. To do this, the solvers repeatedly call F = ODEFILE(T,Y) where argument T is a scalar, Y is a column vector, and output F is expected to be a column vector of the same length. Note that the ODE file must accept the arguments T and Y, although it does not have to use them. In its simplest form, an ODE file can be coded as

```
function F = odefile(t,y)
F = < Insert a function of t and/or y here. >;
```

As described in the User's Guide, the ODE Suite solvers are capable of using additional information coded in the ODE file. In this more general usage, an ODE file is expected to respond to the arguments ODEFILE(T,Y,FLAG,P1,P2,...) where T and Y are the integration variables, FLAG is a lower case string indicating the type of information that the ODE file should return, and P1,P2,... are any additional parameters that the problem requires. The currently supported flags are

```
FLAGS          RETURN VALUES
'' (empty) - F(t,y)
'init'       - default TSPAN, Y0 and OPTIONS for this problem
'jacobian' - Jacobian matrix J(t,y) = dF/dy
'jpattern' - matrix showing the Jacobian sparsity pattern
'mass'        - mass matrix M(t) for solving M(t)*y' = F(t,y)
'events'    - information for zero-crossing location
```

See also RIGIDODE, VDPODE, BRUSSODE, FEM1ODE, and BALLODE M-files for examples of FLAG usage. The VDPODE, BRUSSODE and FEM1ODE examples illustrate the use of additional ODE file parameters, and are also examples of 'vectorized' ODE files (an option described in ODESET).

Below is a template that illustrates how you might code an extended ODE file that uses two additional input parameters. This template uses subfunctions. Note that it is not typical to include all of the cases shown below. For example, 'jacobian' information is used for evaluating Jacobians analytically, and 'jpattern' information is used for generating Jacobians numerically.

```
function varargout = odefile(t,y,flag,p1,p2)

switch flag
case ''                                   % Return dy/dt = f(t,y).
  varargout{1} = f(t,y,p1,p2);
case 'init'                               % Return default [tspan,y0,options].
  [varargout{1:3}] = init(p1,p2);
case 'jacobian'                           % Return Jacobian matrix df/dy.
  varargout{1} = jacobian(t,y,p1,p2);
```

```
case 'jpattern'                          % Return sparsity pattern matrix S.
  varargout{1} = jpattern(t,y,p1,p2);
case 'mass'                              % Return mass matrix M(t) or M.
  varargout{1} = mass(t,y,p1,p2);
case 'events'                            % Return [value,isterminal,direction].
  [varargout{1:3}] = events(t,y,p1,p2);
otherwise
  error(['Unknown flag ''' flag '''.']);
end

% ------------------------------------------------------------------

function dydt = f(t,y,p1,p2)
dydt = < Insert a function of t and/or y, p1, and p2 here. >

% ------------------------------------------------------------------

function [tspan,y0,options] = init(p1,p2)
tspan = < Insert tspan here. >;
y0 = < Insert y0 here. >;
options = < Insert options = odeset(...) or [] here. >;

% ------------------------------------------------------------------

function dfdy = jacobian(t,y,p1,p2)
dfdy = < Insert Jacobian matrix here. >;

% ------------------------------------------------------------------

function S = jpattern(t,y,p1,p2)
S = < Insert Jacobian matrix sparsity pattern here. >;

% ------------------------------------------------------------------

function M = mass(t,y,p1,p2)
M = < Insert mass matrix here. >;

% ------------------------------------------------------------------

function [value,isterminal,direction] = events(t,y,p1,p2)
value = < Insert event function vector here. >
isterminal = < Insert logical ISTERMINAL vector here.>;
direction = < Insert DIRECTION vector here.>;
```

用户可以将 ODE 文件的内容剪切和粘贴到一个独立的文件中。然后在该文件中进行编辑操作，去掉那些用户不需要的内容。并在说明处插入需要给定的信息。其中进行 ODE 系统的定义是必须的。

有以下几点说明：

1，ODE 文件必须从 ODE 求解器中接受 t 和 y 向量，而且必须返回一个和 y 具有相同长度的列向量。其中可选的输入参数 flag 可以决定 ODE 文件返回的输出的类型，例如：质量矩阵、Jacobian 矩阵等。

2，求解器在不同的时间步反复调用 ODE 文件，计算相应的微分方程。因此用户必须首先定义准备求解的 ODE 系统。

3，参数 switch 用来决定需要的输出类型，这样 ODE 文件就可以将相应的信息传递给求解器。

4，在缺省的初始条件('init')项中，ODE 文件为求解器返回一些基本信息，例如，时间跨度、初始条件以及选项等。如果用户忽略这项，就需要在后面计算时，在命令行中输入所有这些基本信息。

5，在 'Jacobian' 项中，ODE 文件为求解器返回一个 Jacobian 矩阵。仅当用户需要提高 ode15s 和 ode23s 等刚性(stiff)求解器的求解性能时，才需要输入这项。

6，在 'jpattern' 项中，ODE 文件为求解器返回一个稀疏矩阵。仅当用户需要采用数值方法为一个刚性(stiff)求解器生成稀疏 Jacobian 矩阵时，才需要输入这项。

7，在 'mass' 项中，ODE 文件为求解器返回一个质量矩阵。仅当用户需要求解形如 $M(t,y)y' = F(t, y)$ 的方程时，才需要输入这项。

8，在 'events' 项中，ODE 文件为求解器返回完成定位事件所需要的数值。如果 Events 属性被设置为 1，表示 ODE 求解器将检查 event 向量中任何转换为 0 的元素。

9，任何一个未定义的 flag 都会产生一个错误信息。

举例：

van der Pol 方程 $y_1'' - \mu(1 - y_1^2)y_1' + y_1 = 0$ 等价于如下所示的耦合一阶微分方程系统：

$y_1' = y_2$

$y_2' = \mu(1 - y_1^2)y_2 - y_1$。

该方程系统可以利用如下所示的 M 文件进行定义：

```
function out1 = vdp1(t,y)
out1 = [y(2); (1-y(1)^2)*y(2) - y(1)];
```

对于给定的初始条件 y(1) = 2 与 y(2) = 0，和时间区间[0 20]，为了求解 van der Pol 方程，在 MATLAB 命令窗口中执行[t,y] = ode45('vdp1',[0 20],[2; 0])。

为了直观起见，在 MATLAB 窗口中调用函数 plot(t,y(:,1),'-',t,y(:,2),'-.')，将计算结果在窗口中显示出来，如图 9-3 所示。

用户也可以完全在 M 文件中实现微分方程的求解过程。此时，需要用如下所示的内容来修改 vdp1.m 文件：

```
function [out1,out2,out3] = vdp1(t,y,flag)
if nargin < 3 | isempty(flag)
   out1 = [y(1).*(1-y(2).^2)-y(2); y(1)];
else
   switch(flag)
      case 'init'                    % Return tspan, y0, and options
         out1 = [0 20];
         out2 = [2; 0];
         out3 = [];
   otherwise
      error(['Unknown request ''' flag '''.']);
   end
end
```

此时用户只需执行[T,Y] = ode23('vdp1')就可以进行微分方程的求解运算，而不需在命令行中输入任何其他参数。

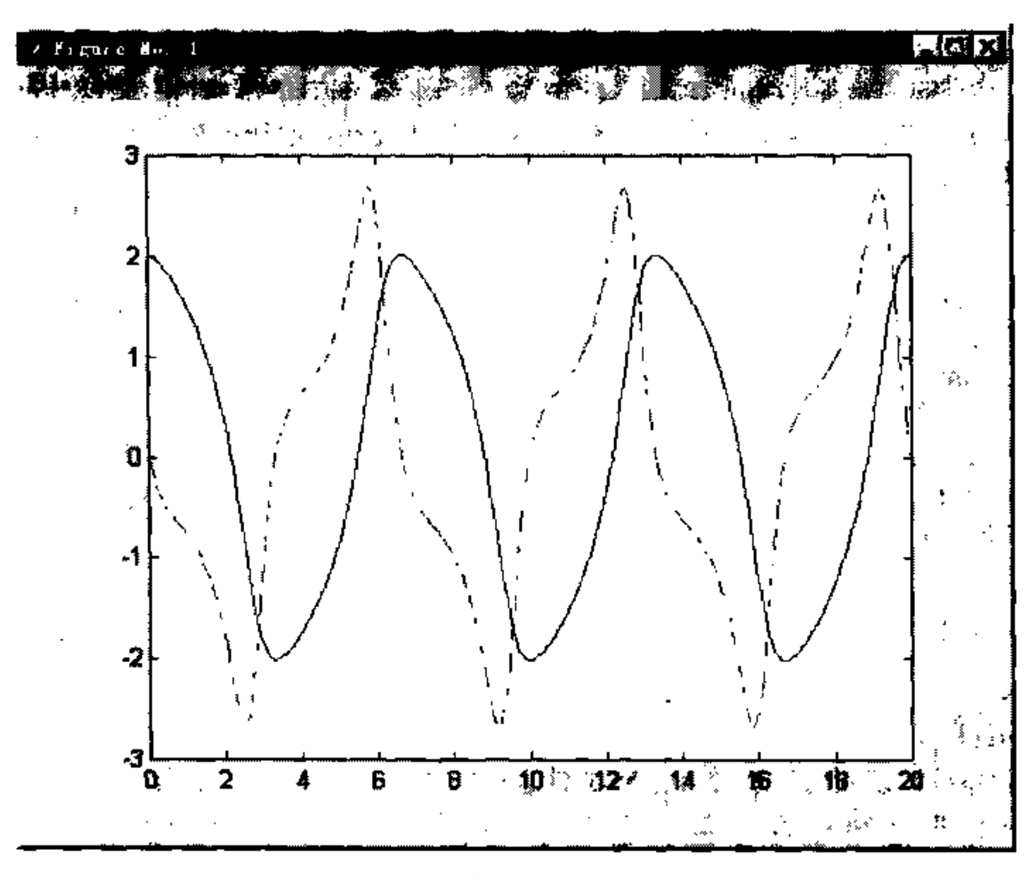


图 9-3 微分方程解的曲线

5. 获取 options 结构的属性

名称：odeget

获取由函数 odeset 所创建的 options 结构的属性。

语法：该函数有如下两种表达形式：

- o = odeget(options,'name')
- o = odeget(options,'name',default)

描述：o = odeget(options,'name')从积分选项结构 options 中返回字符串 same 所指定的属性的取值。如果该属性在 options 结构中没有给出，则返回一个空矩阵[ ]。

o = odeget(options,'name',default)如果选项结构 options 中字符串 name 所指定的属性没有被指定，则返回 o = default。否则，返回 o = on。

举例：

首先利用函数 options1 = odeset('AbsTol',1e-5,'BDF','on','InitialStep',0.5,'JPattern','on')创建一个选项结构，结果如下所示：

```
options1 =
            AbsTol: 1.0000e-005
               BDF: 'on'
            Events: []
       InitialStep: 0.5000
          Jacobian: []
```

```
     JConstant: []
      JPattern: 'on'
          Mass: []
  MassConstant: []
      MaxOrder: []
       MaxStep: []
   NormControl: []
     OutputFcn: []
     OutputSel: []
        Refine: []
        RelTol: []
         Stats: []
    Vectorized: []
```

然后调用函数 odeget(options1,'AbsTol')，返回 AbsTol 属性的取值，结果为：ans =1.0000e-005。

若输入函数 odeget(options1,'RelTol')，则得到：ans = [ ]。

若输入函数 o = odeget(options1,'Mass','default')，则得到：o = default。

若输入函数 o = odeget(options1,'JPattern','default')，则得到：o = on。

6. *创建或修改 options 结构*

名称：odeset

创建或修改 ODE 求解器所需的 options 结构。

语法：该函数有如下几种表达形式：

- options = odeset('name1',value1,'name2',value2,...)
- options = odeset(oldopts,'name1',value1,...)
- options = odeset(oldopts,newopts)
- odeset

描述：函数 odeset 允许用户调整 ODE 求解器中的积分参数。

options = odeset('name1',value1,'name2',value2,...)创建一个积分选项结构。在该结构中，每个定义的属性(name1，name2，…)都被赋予一个特定的值(value1，value2，…)。对于没有给定值的那些属性，odeset 函数会自动赋予其一个空矩阵[ ]。

options = odeset(oldopts,'name1',value1,...)用给定的值来替换已经存在的积分选项结构。

options = odeset(oldopts,newopts)通过把用户提供的选项结构 newopts 和已经存在的选项结构 oldopts 进行结合运算，从而得到一个新的选项结构。

例如，假设已经存在的选项结构为：

oldopts

F	1	[ ]	4	's'	's'	[ ]	[ ]	[ ]

用户提供的新的选项结构为：

newopts

T	3	F	[ ]	' '	[ ]	[ ]	[ ]	[ ]

执行函数 odeset(oldopts,newopts)得到的新的选项结构为：

odeset(oldopts,newopts)

T	3	F	4	' '	's'	[ ]	[ ]	[ ]

单独执行函数 odeset 将显示选项结构中所有属性名及其可能的取值。如下所示：

```
odeset
          AbsTol: [ positive scalar or vector {1e-6} ]
             BDF: [ on | {off} ]
          Events: [ on | {off} ]
     InitialStep: [ positive scalar ]
        Jacobian: [ on | {off} ]
       JConstant: [ on | {off} ]
        JPattern: [ on | {off} ]
            Mass: [ on | {off} ]
    MassConstant: [ on | off ]
        MaxOrder: [ 1 | 2 | 3 | 4 | {5} ]
         MaxStep: [ positive scalar ]
     NormControl: [ on | {off} ]
       OutputFcn: [ string ]
       OutputSel: [ vector of integers ]
          Refine: [ positive integer ]
          RelTol: [ positive scalar {1e-3} ]
           Stats: [ on | {off} ]
      Vectorized: [ on | {off} ]
```

选项结构中的某一项属性是否可用依赖于所使用的 ODE 求解器。但从总体来说，所有属性可以分为 7 大类：

1、误差允许值属性，如表 9-3 中所示。
2、求解器输出属性，如表 9-4 中所示。
3、Jacobian 矩阵属性，如表 9-5 中所示。
4、事件位置属性，如表 9-6 中所示。
5、质量矩阵属性，如表 9-7 中所示。
6、步长属性，如表 9-8 中所示。
7、ode15s 属性，如表 9-9 中所示。

表 9-3　误差允许值属性

属性	取值	描述
RelTol	正的标量值，缺省为 1e-3。	该属性给出了适用于求解向量所有分量的相对误差允许值。
AbsTol	正的标量值或向量，缺省为 1e-6。	绝对误差允许值。如果是标量，则该允许值适用于求解向量的所有分量。否则，该允许值只能适用于相应的分量。

表 9-4　　求解器输出属性

属性	取值	描述
OutputFcn	字符串	安装型输出函数名。例如 odeplot、odephas2、odephas3 和 odeprint。ODE 求解器在进行积分之前调用函数 outputfcn(TSPAN,Y0,'init')来初始化输出函数。接下来，在计算得到每一个输出点(T,Y)之后，调用函数 status = outputfcn(T,Y)。如果积分应该被终止(例如按下 STOP 按钮)，则返回 1，否则返回 0。当积分全部完成时，求解器调用函数 outputfcn([],[],'done')。
OutputSel	索引向量	指定求解向量的哪个分量将被传递到输出函数。
Refine	正整数	通过改变因子 n 来增加输出点数目，从而得到更加光滑的输出结果。对于大多数求解器，Refine 属性的缺省值为 1。但是在 ode45 求解器中，为了补偿大步长引起的缺陷，将 Refine 属性的缺省值取为 4。如果仅仅想观察 ode45 所选择的步长，可令 Refine 取 1。
Stats	On 或 off，缺省为 off。	指定是否显示积分计算所用的时间。

表 9-5　　Jacobian 矩阵属性(适用于 ode15s 和 ode23s)

属性	取值	描述
Jacobian	On 或 off，缺省为 off。	取 On 时，表示 ODE 文件通过返回 $\partial F/\partial y$ 来对(t,y,'jacobian')进行响应。
Jconstant	On 或 off，缺省为 off。	取 On 时，表示 Jacobian 矩阵 $\partial F/\partial y$ 为常数。
Jpattern	On 或 off，缺省为 off。	取 On 时，表示 ODE 文件通过返回一个稀疏矩阵来对([],[],'jpattern')进行响应。
Vectorized	On 或 off，缺省为 off。	取 On 时，表示 ODE 文件 F(t,y)已经被向量化，从而使得运行函数 F(t,[y1 y2 ...])时，返回[F(t,y1) F(t,y2) ...]，换句话说，该 ODE 文件可以将一个列向量数组传递给求解器。

表 9-6　　事件位置属性

属性	取值	描述
Events	On 或 off，缺省为 off。	取 On 时，命令求解器定位事件。

表 9-7　　质量矩阵属性(适用于 ode15s 和 ode23s)

属性	取值	描述
Mass	没有质量矩阵，或质量矩阵为 M、M(t)或 M(t,y)，缺省为没有质量矩阵	说明 ODE 文件是否返回一个质量矩阵。
MassSingular	Yes 或 no 或 maybe，缺省为 maybe。	说明质量矩阵是否是奇异的。

表 9-8　　步长属性

属性	取值	描述
MaxStep	正的标量值	指定求解器使用的最大步长值。
InitialStep	正的标量值	给定一个建议的初始步长。求解器首先采用该步长进行计算，但如果步长太大，则会产生一个错误结果，此时求解器会自动使用一个缩短了的步长进行计算。

表 9-9　　ode15s 属性(只适用于 ode15s 求解器)

属性	取值	描述
MaxOrder	1、2、3、4 或 5，缺省值为 5。	公式中使用的最大阶次。
BDF	On 或 off，缺省为 off。	取 On 时，表示使用后微分公式(BDFa)，而不是使用缺省的数值微分公式(NDFs)。

举例：

在 MATLAB 命令窗口中输入 options1 = odeset('AbsTol',1e-5,'BDF','on','InitialStep',0.5)，可得：

```
options1 =
          AbsTol: 1.0000e-005
             BDF: 'on'
          Events: []
     InitialStep: 0.5000
        Jacobian: []
       JConstant: []
        JPattern: []
            Mass: []
    MassConstant: []
        MaxOrder: []
         MaxStep: []
     NormControl: []
       OutputFcn: []
       OutputSel: []
          Refine: []
          RelTol: []
           Stats: []
      Vectorized: []
```

此时若执行函数 options2 = odeset(options1,'JPattern','on','MaxOrder',2,'MaxStep',10) 对已经定义的选项结构进行修改，可得：

```
options2 =
          AbsTol: 1.0000e-005
             BDF: 'on'
          Events: []
     InitialStep: 0.5000
        Jacobian: []
       JConstant: []
        JPattern: 'on'
            Mass: []
    MassConstant: []
        MaxOrder: 2
```

```
        MaxStep: 10
    NormControl: []
      OutputFcn: []
      OutputSel: []
         Refine: []
         RelTol: []
          Stats: []
     Vectorized: []
```

7. 积分的数值解

名称：quad, quad8

积分的数值解。

语法：该函数有如下几种表达形式：

- q = quad('fun',a,b)
- q = quad('fun',a,b,tol)
- q = quad('fun',a,b,tol,trace)
- q = quad('fun',a,b,tol,trace,P1,P2,...)
- q = quad8(...)

描述：函数 quad 和 quad8 均为数值积分函数。两者均采用迭代算法。其中，前者采用自适应 Simpson 法则，后者采用自适应 Newton Cotes 8 panel 法则。

quad 函数的调用格式为：

q = quad('fun',a,b)返回数值积分的结果。而且如果输入的函数为一个向量，则输出的结果也为一个同维向量。

q = quad('fun',a,b,tol)进行反复的积分求解迭代，直到计算得到的相对误差小于 tol 值时迭代停止。

q = quad('fun',a,b,tol,trace) 进行反复的积分求解迭代，直到计算得到的相对误差小于 tol 值时迭代停止。当 trace 取为非零值时，就会将积分过程用图形显示出来。

q = quad('fun',a,b,tol,trace,P1,P2,...)还允许将系数 P1,P2,...直接传递到指定的函数 G=fun(X,P1,P2,...)。

其中字符串 fun 为函数名；a 和 b 用来指定积分区间；tol 为可选项，指定迭代的误差限，缺省值为 1.e-3；trace 也为可选项，当其不为 0 时，将会绘制出所采用的积分点的图形。

函数 quad8 的调用格式与 quad 类似。

举例：

下面的例子将利用函数 quad 求螺旋线的长度。

已知螺旋线 x(t) = sin(t)，y(t) = cos(t)，z(t) = t∈ (0, 2π)。

1，由弧长公式可知，该曲线长度为：$L=\int_0^{2\pi}\sqrt{\cos(t)^2+4\sin(2t)^2+1}dt$，用文件 LEN.m 生成该函数。LEN.m 文件的内容为：

```
function f=LEN(t)
f=sqrt(cos(t).^2+4*sin(2*t).^2+1);
plot3(sin(2*t),cos(t),t)
```

2，调用函数 quad 进行积分计算，命令为：

```
length = quad('LEN',0,2*pi)
```

结果为：

```
length =

    11.4609
```

所以曲线的长度大约为 11.5。

8. 向量化表示

名称：vectorize

向量化表示。

语法：该函数有如下两种表达形式：

- vectorize(string)
- vectorize(function)

描述：vectorize(string)在 'string' 中任意的 '^'、'*' 或 '/' 之前插入一个 '.'。结果是一个字符串。

vectorize(function)如果输入参数是一个内置函数块，将得到 function 的向量形式。

举例：

例如执行 vectorize('26*78^93/356')时，将得到：

```
ans =

26.*78.^93./356
```

如果例如执行 vectorize('matlab')时，则得到：

```
ans =

matlab
```

第十章 稀疏矩阵函数

含有大量 0 的矩阵，称为稀疏矩阵。在工程问题中，稀疏矩阵是很常见的。比如，在有限元计算中，当由小矩阵(单元刚度矩阵)集成为大矩阵(总体刚度矩阵)时，将会产生大量的 0 元素，而且形成的矩阵越大，矩阵越稀疏。如果按照普通矩阵对待稀疏矩阵，0 元素将占据大量的存储空间和内存空间，从而影响运行速度。工程师门为了提高稀疏矩阵的运算效率，采用只存储非 0 元素的方法，例如采用矩阵的二维等带宽存储、一维变带宽存储等方法进行矩阵的存储和运算。

由于 0 元素占用和非 0 元素同样的空间，对 0 元素运算将花费很多时间，在 MATLAB 里，只存储非 0 元素的值和下标，这对于严重稀疏的矩阵，将大大减少内存占用；运算时，只对非 0 元素进行运算。

在 MATLAB 内部，采用三个向量来存储稀疏矩阵：第一个用于存储非 0 元素的值、第二个向量存储非 0 元素的行下标，第三个向量存储每列的第一个非 0 元素行下标。

假定一个 m×n 的实矩阵中有 nnz 个非 0 元素，则第一个向量的长度为 nnz，用来存储非 0 元素的值，第二个向量的长度也为 nnz，第三个向量的长度为 n。由于一个浮点数占 8 个字节，一个整数占 4 个字节，所以，整个稀疏矩阵所占的字节数为：

$8\times nnz+4\times(nnz+n)$

如果为复矩阵，MATLAB 将采用第四个向量来存储非 0 元素的虚部。

本章主要介绍 MATLAB 中稀疏矩阵的生成和进行稀疏矩阵运算所需要的函数。

10.1 基本稀疏矩阵

1. *稀疏带状矩阵*

名称：spdiags

生成稀疏带状矩阵。

语法：该函数有如下几种表达形式：

- [B,d] = spdiags(A)
- B = spdiags(A,d)
- A = spdiags(B,d,A)
- A = spdiags(B,d,m,n)

描述：在 MATLAB 里，不会自动生成稀疏矩阵，只有根据矩阵中非 0 元素的多少，确定是否把矩阵定义为稀疏矩阵。为了说明的方便，先定义矩阵密度的概念。矩阵密度为矩阵中非 0 元素的数目和元素总和之比。矩阵密度反应了矩阵的稀疏程度。密度越小，表明矩阵越稀疏。

在实际工程应用中，矩阵元素集中在对角线附近是很常见的，这种矩阵被称为带状矩

阵。在这种情况下，采用函数 spdiags 可以方便地生成稀疏矩阵。其调用格式为：

[B,d] = spdiags(A)返回 m×n 阶矩阵 A 中所有的非零对角线元素。其中 B 为 min(m,n)×p 阶矩阵，其列向量为 A 的 p 阶非零对角线元素；d 为长度为 p 的向量，其整数分量指定矩阵 A 的对角线。

B = spdiags(A,d)返回一个由 d 所指定的对角线矩阵。

A = spdiags(B,d,A)用矩阵 B 中的列代替由 d 所指定的对角线元素，并返回一个稀疏矩阵。

A = spdiags(B,d,m,n)创建一个 m×n 阶稀疏矩阵。

举例：

例如对于如下所示的一个任意稀疏矩阵 A：

```
A =
    11     0    13     0     0
    41    22     0    24     0
     0    52    33     0    17
     0     0    63    44     0
     0     0     0    74    14
     0     0     0     0    23
```

调用函数[B,d] = spdiags(A)，可以得到其对角线元素所组成的矩阵 B 和相应的向量 b，结果如下所示：

```
B =
    41    11     0
    52    22     0
    63    33    13
    74    44    24
    23    14    17
d =
    -1
     0
     2
```

若调用函数 B = spdiags(A,[-1 0])，则可得到：

```
B =
    41    11
    52    22
    63    33
    74    44
    23    14
```

再如对于如下所示的矩阵：

```
B =
     1    -2     1
```

```
     1    -2     1
     1    -2     1
     1    -2     1
     1    -2     1
     1    -2     1
     1    -2     1
     1    -2     1
     1    -2     1
     1    -2     1
```

调用函数 A = spdiags(B, -1:1, 10, 10)，可以得到：

```
A =
   (1,1)       -2
   (2,1)        1
   (1,2)        1
   (2,2)       -2
   (3,2)        1
   (2,3)        1
   (3,3)       -2
   (4,3)        1
   (3,4)        1
   (4,4)       -2
   (5,4)        1
   (4,5)        1
   (5,5)       -2
   (6,5)        1
   (5,6)        1
   (6,6)       -2
   (7,6)        1
   (6,7)        1
   (7,7)       -2
   (8,7)        1
   (7,8)        1
   (8,8)       -2
   (9,8)        1
   (8,9)        1
   (9,9)       -2
  (10,9)        1
   (9,10)       1
  (10,10)      -2
```

其对应的稀疏矩阵为：

```
A =
    -2     1     0     0     0     0     0     0     0     0
     1    -2     1     0     0     0     0     0     0     0
     0     1    -2     1     0     0     0     0     0     0
     0     0     1    -2     1     0     0     0     0     0
     0     0     0     1    -2     1     0     0     0     0
     0     0     0     0     1    -2     1     0     0     0
     0     0     0     0     0     1    -2     1     0     0
     0     0     0     0     0     0     1    -2     1     0
     0     0     0     0     0     0     0     1    -2     1
     0     0     0     0     0     0     0     0     1     2
```

2. 单位稀疏矩阵

名称：speye

生成单位稀疏矩阵。

语法：该函数有如下两种表达式：

- S = speye(m,n)
- S = speye(n)

描述：S = speye(m,n)返回一个主对角线元素取值为 1 的 m×n 阶单位稀疏矩阵。

S = speye(n)是 speye(n,n)的缩写形式。

举例：

例如调用函数 S = speye(15,10)，可得到一个 15×10 阶单位稀疏矩阵：

```
S =
     1     0     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

若调用函数 S = speye(6)，可得一个 6×6 阶单位稀疏矩阵：

S =

1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1

3. 随机稀疏矩阵

名称：sprand

均匀分布的随机稀疏矩阵。

语法：该函数有如下几种表达形式：

- R = sprand(S)
- R = sprand(m,n,density)
- R = sprand(m,n,density,rc)

描述：R = sprand(S)返回一个和矩阵 S 具有相同密度，但其中非 0 项的取值呈均匀随机分布的稀疏矩阵。

R = sprand(m,n,density)返回一个 m×n 阶的均匀随机分布稀疏矩阵。其中参数 density 用来指定稀疏矩阵的密度，它的取值范围为[0 1]。

R = sprand(m,n,density,rc)返回一个 m×n 阶的均匀随机分布稀疏矩阵。其中参数 density 用来指定稀疏矩阵的密度，它的取值范围为[0 1]。参数 rc 用来指定矩阵的条件数。

举例：

例如对于 6×6 阶单位稀疏矩阵 S：

S =

1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1

调用函数 R = sprand(S)，得到：

R =

(1,1)	0.9501
(2,2)	0.2311
(3,3)	0.6068
(4,4)	0.4860
(5,5)	0.8913
(6,6)	0.7621

其对应的稀疏矩阵为：

R =

```
0.9501      0       0       0       0       0
     0  0.2311      0       0       0       0
     0       0  0.6068      0       0       0
     0       0       0  0.4860      0       0
     0       0       0       0  0.8913      0
     0       0       0       0       0  0.7621
```

调用函数 R = sprand(6,6,0.4)，可以得到：

```
R =
   (1,1)        0.1988
   (5,1)        0.7939
   (3,2)        0.0099
   (1,3)        0.6252
   (3,3)        0.4199
   (5,3)        0.9200
   (6,3)        0.3678
   (2,4)        0.3759
   (6,4)        0.6208
   (1,5)        0.7334
   (3,6)        0.7537
   (5,6)        0.8447
```

其对应的稀疏矩阵为：

```
R =
0.1988      0  0.6252       0  0.7334       0
     0      0       0  0.3759       0       0
     0  0.0099  0.4199      0       0  0.7537
     0      0       0       0       0       0
0.7939      0  0.9200       0       0  0.8447
     0      0  0.3678  0.6208       0       0
```

若调用函数 R = sprand(6,6,0.4,[1 2 3 1])，可以得到：

```
R =
   (2,1)        3.0000
   (1,2)        0.1965
   (3,2)       -0.3294
   (4,2)       -1.3451
   (5,2)        0.8697
   (6,2)       -0.6642
   (1,3)       -0.0335
   (3,3)       -0.5266
   (4,3)        0.2295
```

```
(5,3)      -0.5440
(6,3)       1.1026
(1,4)       0.0601
(3,4)       0.7837
(4,4)      -0.4111
(6,4)       0.4617
(1,5)      -0.0053
(4,5)       0.0361
(5,5)      -0.0293
(6,5)       0.0328
```

4. 正态分布的随机稀疏矩阵

名称：sprandn

生成正态分布的随机稀疏矩阵。

语法：该函数有如下几种表达形式：

- R = sprandn(S)
- R = sprandn(m,n,density)
- R = sprandn(m,n,density,rc)

描述：R = sprandn(S)返回一个和矩阵 S 具有相同密度，但其中非 0 项的取值呈正态随机分布(非 0 元素的平均值为 0，方差为 1)的稀疏矩阵。

R = sprandn(m,n,density)返回一个 m×n 阶的正态随机分布稀疏矩阵。其中参数 density 用来指定稀疏矩阵的密度，它的取值范围为[0 1]。

R = sprandn(m,n,density,rc)返回一个 m×n 阶的正态随机分布稀疏矩阵。其中参数 density 用来指定稀疏矩阵的密度，它的取值范围为[0 1]。参数 rc 用来指定矩阵的条件数。

举例：

例如对于 6×6 阶单位稀疏矩阵 S：

```
S =
     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
```

调用函数 R = sprand(S)，得到：

```
R =
   (1,1)        0.3527
   (2,2)        0.1879
   (3,3)        0.4906
   (4,4)        0.4093
   (5,5)        0.4635
```

```
    (6,6)        0.6109
```

其对应的稀疏矩阵为：

```
R =
0.3527      0       0       0       0       0
     0  0.1879      0       0       0       0
     0      0  0.4906       0       0       0
     0      0       0  0.4093       0       0
     0      0       0       0  0.4635       0
     0      0       0       0       0  0.6109
```

调用函数 R = sprandn(6,6,0.4)，可以得到：

```
R =
    (4,1)        1.1892
    (2,2)       -1.6656
    (3,2)        0.2877
    (1,3)       -0.4326
    (3,3)       -1.1465
    (4,3)       -0.0376
    (2,4)        0.1253
    (6,4)        0.1746
    (3,5)        1.1909
    (4,5)        0.3273
```

其对应的稀疏矩阵为：

```
R =
     0       0  -0.4326       0       0       0
     0 -1.6656   0.2877  0.1253       0       0
     0       0  -1.1465       0  1.1909       0
1.1892       0  -0.0376       0  0.3273       0
     0       0        0       0       0       0
     0       0        0  0.1746       0       0
```

若调用函数 R = sprandn(6,6,0.4,[1 2 3 1])，可以得到：

```
R =
    (4,1)        0.7121
    (2,2)        0.1899
    (5,2)        0.2789
    (6,2)        0.0839
    (2,3)        0.5121
    (5,3)        0.7521
    (6,3)        0.2262
    (2,4)       -0.3871
```

```
    (4,4)        0.5362
    (6,4)        0.8763
    (1,5)        2.7210
    (3,5)        1.2635
    (2,6)       -0.1159
    (4,6)       -1.7904
    (6,6)        0.2625
```

5. 稀疏对称随机矩阵

名称：sprandsym

生成稀疏对称随机矩阵。

语法：该函数有如下几种表达形式：

- R = sprandsym(S)
- R = sprandsym(n,density)
- R = sprandsym(n,density,rc)
- R = sprandsym(n,density,rc,kind)

描述：R = sprandsym(S)返回一个对称随机矩阵。其下三角和对角线部分和矩阵 S 具有相同的结构。而且该矩阵中元素呈正态随机分布，平均值为 0，方差为 1。

R = sprandsym(n,density)返回一个 n×n 阶的对称随机稀疏矩阵。其中参数 density 用来指定稀疏矩阵的密度，它的取值范围为[0 1]。

R = sprandsym(n,density,rc)返回一个 n×n 阶的对称随机稀疏矩阵。其中参数 density 用来指定稀疏矩阵的密度，它的取值范围为[0 1]。参数 rc 用来指定矩阵的条件数。

R = sprandsym(n,density,rc,kind)返回一个正定矩阵。其中参数 density 用来指定矩阵的密度，它的取值范围为[0 1]。参数 rc 用来指定矩阵的条件数。参数 kind 有如下几种情况：

取 1，表示通过对一个正定对角矩阵进行随机 Jacobi 旋转来得到矩阵 R。而且 R 含有预期的精确条件数。

取 2，表示通过对向量外积平移来得到矩阵 R。此时 R 含有预期的近似条件数。

取 3，表示要生成一个和 S 具有相同结构的矩阵 R。并且 R 的近似条件数为 1/rc。

举例：

例如任意一个 4×4 阶矩阵 S：

```
S =
     1     3     0     0
     5     6     7     0
     0     4     5     0
     0     0     7     2
```

调用函数 R = sprandsym(S)，得到：

```
R =
    (1,1)       -1.2025
    (2,1)        0.2573
    (1,2)        0.2573
```

```
    (2,2)       -0.0198
    (3,2)       -1.0565
    (2,3)       -1.0565
    (3,3)       -0.1567
    (4,3)        1.4151
    (3,4)        1.4151
    (4,4)       -1.6041
```

其对应的稀疏矩阵为：

```
R =
-1.2025    0.2573         0         0
 0.2573   -0.0198   -1.0565    1.4151
      0   -1.0565   -0.1567         0
      0         0    1.4151   -1.6041
```

调用函数 R = sprandsym(6,0.4)，可以得到：

```
R =
    (2,1)        0.2193
    (3,1)        0.5287
    (4,1)       -0.0592
    (5,1)       -0.9219
    (1,2)        0.2193
    (3,2)        0.6145
    (6,2)       -3.1813
    (1,3)        0.5287
    (2,3)        0.6145
    (1,4)       -0.0592
    (1,5)       -0.9219
    (5,5)       -0.8051
    (2,6)       -3.1813
```

其对应的稀疏矩阵为：

```
R =
      0    0.2193    0.5287   -0.0592   -0.9219         0
 0.2193         0    0.6145         0         0   -3.1813
 0.5287    0.6145         0         0         0         0
-0.0592         0         0         0         0         0
-0.9219         0         0         0   -0.8051         0
      0   -3.1813         0         0         0         0
```

若调用函数 R = sprandsym(6,0.4, [1 2 3 0 2 4])，可以得到：

```
R =
    (1,1)        1.4531
```

```
(2,1)      0.0763
(3,1)      0.2745
(5,1)      0.6567
(6,1)      0.0568
(1,2)      0.0763
(2,2)      2.1363
(3,2)     -0.0303
(5,2)     -0.0724
(6,2)      0.5150
(1,3)      0.2745
(2,3)     -0.0303
(3,3)      2.5737
(5,3)      0.4103
(1,5)      0.6567
(2,5)     -0.0724
(3,5)      0.4103
(5,5)      1.9816
(1,6)      0.0568
(2,6)      0.5150
(6,6)      3.8553
```

10.2 满阵和稀疏矩阵的转换

1. 寻找非 0 元素的下标和值

名称：find

寻找非 0 元素的下标和值。

语法：该函数有如下几种表达形式：

- k = find(x)
- [i,j] = find(X)
- [i,j,v] = find(X)

描述：k = find(x)返回数组 x 中指向非 0 元素的下标。如果 x 中没有非 0 元素，则返回一个空矩阵。

[i,j] = find(X)返回矩阵 X 中非 0 元素的行号 i 和列号 j。

[i,j,v] = find(X)返回一个包含矩阵 X 中非 0 元素的列向量 v。同时返回相应的非 0 元素的行号 i 和列号 j。

举例：

例如对于如下所示的一个任意的列向量：

x =

```
    11
     0
    33
     0
    55
```

调用函数 k = find(x)，可以得到由 x 中非 0 元素的下标所组成的向量：

```
k =
     1
     3
     5
```

若调用函数 m = find(x == 0)，则可得：

```
m =
     2
     4
```

若调用函数 n = find(0 < x & x < 10*pi)，则可得：

```
n =
     1
```

再如对于如下所示的 3 阶 magic 矩阵：

```
M =
     8     1     6
     3     5     7
     4     9     2
```

调用函数[i,j,v] = find(M > 6)

```
i =
     1
     3
     2
j =
     1
     2
     3
v =
     1
     1
     1
```

2. 稀疏矩阵转换为满阵

名称：full

把稀疏矩阵转换为满阵。

语法：A = full(S)

描述：A = full(S)把一个稀疏矩阵转换为一个满阵。如果 S 已经是一个满阵，则返回的矩阵与矩阵 S 相同。

举例：

首先利用函数 S = sparse(rand(10,10) < 2/3)生成一个稀疏矩阵，可得：

```
S =
   (4,1)     1
   (5,1)     1
   (6,1)     1
   (7,1)     1
   (1,2)     1
   (3,2)     1
   (4,2)     1
   (6,2)     1
   (8,2)     1
   (9,2)     1
  (10,2)     1
   (1,3)     1
   (2,3)     1
   (4,3)     1
   (5,3)     1
   (7,3)     1
   (8,3)     1
   (9,3)     1
  (10,3)     1
   (1,4)     1
   (2,4)     1
   (3,4)     1
   (4,4)     1
   (6,4)     1
   (7,4)     1
   (8,4)     1
   (9,4)     1
  (10,4)     1
   (1,5)     1
   (2,5)     1
   (3,5)     1
   (5,5)     1
   (6,5)     1
   (7,5)     1
   (1,6)     1
   (2,6)     1
   (3,6)     1
   (4,6)     1
   (6,6)     1
   (8,6)     1
   (1,7)     1
   (3,7)     1
   (4,7)     1
   (5,7)     1
   (6,7)     1
   (7,7)     1
  (10,7)     1
   (1,8)     1
   (3,8)     1
   (4,8)     1
   (5,8)     1
   (6,8)     1
   (9,8)     1
  (10,8)     1
   (1,9)     1
   (2,9)     1
   (3,9)     1
   (5,9)     1
   (6,9)     1
   (7,9)     1
   (8,9)     1
   (9,9)     1
   (5,10)    1
   (6,10)    1
   (8,10)    1
   (9,10)    1
  (10,10)    1
```

然后调用函数 A = full(S)，将该稀疏矩阵转换为一个满阵，结果如下所示：

```
A =
     0     1     1     1     1     1     1     1     1     0
     0     0     1     1     1     1     0     0     1     0
     0     1     0     1     1     1     1     1     1     0
     1     1     1     1     0     1     1     1     0     0
     1     0     1     0     1     0     1     1     1     1
     1     1     0     1     1     1     1     1     1     1
     1     0     1     1     1     0     1     0     1     0
```

```
0  1  1  1  0  1  0  0  1  1
0  1  1  1  0  0  0  1  1  1
0  1  1  1  0  0  1  1  0  1
```

3. 生成稀疏矩阵

名称：sparse

生成稀疏矩阵。

语法：该函数有如下几种表达形式：

- S = sparse(A)
- S = sparse(i,j,s,m,n,nzmax)
- S = sparse(i,j,s,m,n)
- S = sparse(i,j,s)
- S = sparse(m,n)

描述：函数 sparse 可以把满阵转换成稀疏矩阵的形式。其调用格式为：

S = sparse(A)把一个满阵 A 转换为稀疏矩阵的形式。

S = sparse(i,j,s,m,n,nzmax)利用向量 i、j 和 s 生成一个包含 nzmax 个非零元素的 m×n 阶稀疏矩阵。而且向量 i、j 和 s 具有相同的长度。其中 i 和 j 分别为非 0 元素的横坐标和纵坐标的下标所组成的向量，s 是对应于下标为(i,j)的元素值，m 和 n 分别为矩阵的行数和列数。nzmax 为稀疏矩阵中非 0 元素的数目。

S = sparse(i,j,s,m,n)利用向量 i、j 和 s 生成一个包含 length(s)个非零元素的 m×n 阶稀疏矩阵。而且向量 i、j 和 s 具有相同的长度。

S = sparse(i,j,s)利用向量 i、j 和 s 生成一个包含 length(s)个非零元素的 max(i)× max(j)阶稀疏矩阵。而且向量 i、j 和 s 具有相同的长度。

S = sparse(m,n)是 sparse([],[],[],m,n,0)的缩写形式。该函数将返回一个元素全部为 0 的 m×n 阶稀疏矩阵。

举例：

例如对于如下所示的一个满阵：

```
A =
    0    0    0    5
    0    2    0    0
    1    3    0    0
    0    0    4    0
```

调用函数 S = sparse(A)可将其转换为一个稀疏矩阵：

```
S =
   (3,1)        1
   (2,2)        2
   (3,2)        3
   (4,3)        4
   (1,4)        5
```

上式中列出了稀疏矩阵中非 0 元素的下标和元素值。

再如，当运行函数 S=sparse([3 2 3 4 1],[1 2 2 3 4],[1 2 3 4 5],4,4)时，将得到：

```
S =
   (3,1)        1
   (2,2)        2
   (3,2)        3
   (4,3)        4
   (1,4)        5
```

执行下面的命令可以生成三对角矩阵：

```
n = 4
D = sparse (1:n,1:n,4*ones(1,n),n,n)
E = sparse (2:n,1:n-1,ones(1,n-1),n,n)
S=E+D-E'
```

结果为：

```
S =
   (1,1)        4
   (2,1)        1
   (1,2)       -1
   (2,2)        4
   (3,2)        1
   (2,3)       -1
   (3,3)        4
   (4,3)        1
   (3,4)       -1
   (4,4)        4
```

表示为矩阵形式为：

```
S =
     4    -1     0     0
     1     4    -1     0
     0     1     4    -1
     0     0     1     4
```

4. 载入稀疏矩阵

名称：spconvert

从外部格式中载入稀疏矩阵。

语法：S = spconvert(D)

描述：命令 load 和 spconvert 结合使用，可以把在 MATLAB 以外生成的稀疏矩阵的文本文件调入 MATLAB 工作空间。在文本文件中，第一列为矩阵中非 0 元素的行下标，第二列为列下标，第三列为元素值。先用命令将包含稀疏矩阵信息的矩阵调入工作空间，再用函数 spconvert 将其转换为稀疏矩阵。其调用格式为：

S = spconvert(D)

另外，命令save和load也可将矩阵存入MAT文件。用户可以调用FORTRAN程序hbo2mat将Harwell-Boeing格式的稀疏矩阵文件转换为MAT文件格式，从而可以用load命令来调入。

举例：

首先编辑一个ASCII文本文件，为其命名为uphill.dat，其中包括如下所示的数据：

```
1   1   1.000000000000000
1   2   0.500000000000000
2   2   0.333333333333333
1   3   0.333333333333333
2   3   0.250000000000000
3   3   0.200000000000000
1   4   0.250000000000000
2   4   0.200000000000000
3   4   0.166666666666667
4   4   0.142857142857143
4   4   0.000000000000000
```

然后调用函数 load uphill.dat 和 H = spconvert(uphill)，将 uphill.dat 文件中的数据调入MATLAB工作空间，并将其转换为稀疏矩阵。结果如下所示：

```
H =
   (1,1)        1.0000
   (1,2)        0.5000
   (2,2)        0.3333
   (1,3)        0.3333
   (2,3)        0.2500
   (3,3)        0.2000
   (1,4)        0.2500
   (2,4)        0.2000
   (3,4)        0.1667
   (4,4)        0.1429
```

对应的矩阵形式为：

```
H =
    1.0000    0.5000    0.3333    0.2500
         0    0.3333    0.2500    0.2000
         0         0    0.2000    0.1667
         0         0         0    0.1429
```

10.3　稀疏矩阵的非 0 元素操作

1. 非 0 元素的个数

名称：nnz

矩阵中非 0 元素的个数。

语法：n = nnz(X)

描述：n = nnz(X)返回矩阵 X 中非 0 元素的个数。

矩阵的密度定义为 nnz(X)/prod(size(X))。

举例：

首先利用函数 X = sparse(wilkinson(21))生成一个稀疏矩阵，结果如下所示：

```
X =
  Columns 1 through 12
    10     1     0     0     0     0     0     0     0     0     0     0
     1     9     1     0     0     0     0     0     0     0     0     0
     0     1     8     1     0     0     0     0     0     0     0     0
     0     0     1     7     1     0     0     0     0     0     0     0
     0     0     0     1     6     1     0     0     0     0     0     0
     0     0     0     0     1     5     1     0     0     0     0     0
     0     0     0     0     0     1     4     1     0     0     0     0
     0     0     0     0     0     0     1     3     1     0     0     0
     0     0     0     0     0     0     0     1     2     1     0     0
     0     0     0     0     0     0     0     0     1     1     1     0
     0     0     0     0     0     0     0     0     0     1     0     1
     0     0     0     0     0     0     0     0     0     0     1     1
     0     0     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
  Columns 13 through 21
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
```

```
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     1     0     0     0     0     0     0     0     0
     2     1     0     0     0     0     0     0     0
     1     3     1     0     0     0     0     0     0
     0     1     4     1     0     0     0     0     0
     0     0     1     5     1     0     0     0     0
     0     0     0     1     6     1     0     0     0
     0     0     0     0     1     7     1     0     0
     0     0     0     0     0     1     8     1     0
     0     0     0     0     0     0     1     9     1
     0     0     0     0     0     0     0     1    10
```

然后调用函数 n = nnz(X)，计算矩阵中所包含的非 0 元素的个数，结果为：

```
n =
    60
```

2. 非 0 元素

名称：nonzeros

矩阵中非 0 元素。

语法：s = nonzeros(A)

描述：s = nonzeros(A)返回一个由矩阵 A 中的非 0 元素所组成的列向量。

举例：

对于如下所示的一个 15×15 稀疏矩阵：

```
A =
  Columns 1 through 12
     7     1     0     0     0     0     0     0     0     0     0     0
     1     6     1     0     0     0     0     0     0     0     0     0
     0     1     5     1     0     0     0     0     0     0     0     0
     0     0     1     4     1     0     0     0     0     0     0     0
     0     0     0     1     3     1     0     0     0     0     0     0
     0     0     0     0     1     2     1     0     0     0     0     0
     0     0     0     0     0     1     1     1     0     0     0     0
     0     0     0     0     0     0     1     0     1     0     0     0
     0     0     0     0     0     0     0     1     1     1     0     0
     0     0     0     0     0     0     0     0     1     2     1     0
```

```
     0     0     0     0     0     0     0     0     0     1     3     1
     0     0     0     0     0     0     0     0     0     0     1     4
     0     0     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
  Columns 13 through 15
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     1     0     0
     5     1     0
     1     6     1
     0     1     7
```

调用函数 s = nonzeros(A)，可以得到一个由 A 中所有的非 0 元素所组成的列向量，如下所示：

```
s' =
     7     1     1     6     1     1     5     1     1
     4     1     1     3     1     1     2     1     1
     1     1     1     1     1     1     1     1     2
     1     1     3     1     1     4     1     1     5
     1     1     6     1     1     7
```

3. 非 0 元素存储空间数

名称：nzmax

为非 0 元素分配的存储空间数。

语法：n = nzmax(S)

描述：n = nzmax(S)返回为非 0 元素分配的存储空间数。

如果 S 是一个稀疏矩阵，则函数 nzmax(S)返回为 S 中的非 0 元素所分配的存储空间数。

如果 S 是一个满阵，则函数 nzmax(S)等价于函数 prod(size(S))。

举例：

对于如下所示的一个 10×10 阶稀疏矩阵：

S =

```
  Columns 1 through 7
    4.5000    1.0000         0         0         0         0         0
    1.0000    3.5000    1.0000         0         0         0         0
         0    1.0000    2.5000    1.0000         0         0         0
         0         0    1.0000    1.5000    1.0000         0         0
         0         0         0    1.0000    0.5000    1.0000         0
         0         0         0         0    1.0000    0.5000    1.0000
         0         0         0         0         0    1.0000    1.5000
         0         0         0         0         0         0    1.0000
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
  Columns 8 through 10
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
    1.0000         0         0
    2.5000    1.0000         0
    1.0000    3.5000    1.0000
         0    1.0000    4.5000
```

调用函数 n = nzmax(S)可以计算得到矩阵 S 中的非 0 元素所分配的存储空间数。

```
n =
    28.
```

4. 稀疏矩阵存储空间

名称：spalloc

为稀疏矩阵分配的存储空间。

语法：S = spalloc(m,n,nzmax)

描述：S = spalloc(m,n,nzmax)返回一个全部为 0 元素的 m×n 阶稀疏矩阵。

举例：

例如调用函数 S = spalloc(10,8,5)可以得到一个全部为 0 元素的 10×8 阶稀疏矩阵，如下所示：

```
S =
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
```

```
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
```

5. 非 0 元素的计算

名称：spfun

稀疏矩阵中非 0 元素的函数计算。

语法：f = spfun('function',S)

描述：f = spfun('function',S)返回矩阵 S 中的非 0 元素在函数 function 中的取值。其中字符串 function 为函数名(该函数通常在 M 文件中定义)。

举例：

例如对于如下所示的 4×4 阶对角矩阵：

```
S =
     1     0     0     0
     0     2     0     0
     0     0     3     0
     0     0     0     4
```

调用函数 f = spfun('exp',S)，对 S 中的非 0 元素进行指数运算，最后可以得到：

```
f =
   (1,1)        2.7183
   (2,2)        7.3891
   (3,3)       20.0855
   (4,4)       54.5982
```

其对应的矩阵形式为：

```
f =
    2.7183         0         0         0
         0    7.3891         0         0
         0         0   20.0855         0
         0         0         0   54.5982
```

若用户直接调用函数 full(exp(S))，则得到：

```
ans =
    2.7183    1.0000    1.0000    1.0000
    1.0000    7.3891    1.0000    1.0000
    1.0000    1.0000   20.0855    1.0000
    1.0000    1.0000    1.0000   54.5982
```

从上面可以看出，此时函数 full(exp(S))对 S 中的所有元素(包括 0 元素)进行运算。

6. 非 0 元素全部用 1 替换

名称：spones

将稀疏矩阵中的非 0 元素全部用 1 替换。

语法：R = spones(S)

描述：R = spones(S)将稀疏矩阵 S 中的所有非 0 元素都用 1 替换，并返回这个替换后的矩阵。

举例：

例如对于如下所示的一个 7×7 阶稀疏矩阵：

```
S =
      3     1     0     0     0     0     0
      1     2     1     0     0     0     0
      0     1     1     1     0     0     0
      0     0     1     0     1     0     0
      0     0     0     1     1     1     0
      0     0     0     0     1     2     1
      0     0     0     0     0     1     3
```

调用函数 R = spones(S)，可以将 S 中的所有非 0 元素都用 1 替换，并返回这个替换后的矩阵。结果为：

```
R =
      1     1     0     0     0     0     0
      1     1     1     0     0     0     0
      0     1     1     1     0     0     0
      0     0     1     0     1     0     0
      0     0     0     1     1     1     0
      0     0     0     0     1     1     1
      0     0     0     0     0     1     1
```

10.4 稀疏矩阵的可视化

1. 图形表示

名称：spy

稀疏矩阵的图形表示。

语法：该函数有如下几种表达形式：

- spy(S)
- spy(S,markersize)
- spy(S,'LineSpec')
- spy(S,'LineSpec',markersize)

描述：对于庞大的稀疏矩阵，采用图形的方式来查看矩阵中非 0 元素的分布，将会非常直观。MATLAB 中的 spy 函数就提供了这一功能。spy 函数将矩阵中的每个非 0 元素用一个小点来表示，从点的分布可以看出矩阵非零结构。其调用格式为：

spy(S)绘制任意一个矩阵 S 的结构图。其中 S 既可以是一个普通的稀疏矩阵也可以是一个满阵。

spy(S,markersize)用大小为 markersize 的点绘制矩阵 S 的结构图。其中 S 既可以是一个普通的稀疏矩阵也可以是一个满阵。参数 markersize 是一个整数，指定绘制点的尺寸。

spy(S,'LineSpec')用 LineSpec 指定的线型绘制矩阵 S 的结构图。其中 S 既可以是一个普通的稀疏矩阵也可以是一个满阵。LineSpec 为字符串，用来指定线型和颜色。

spy(S,'LineSpec',markersize)用 LineSpec 指定的线型和大小为 markersize 的点绘制矩阵 S 的结构图。其中 S 既可以是一个普通的稀疏矩阵也可以是一个满阵。LineSpec 为字符串，用来指定线型和颜色。参数 markersize 是一个整数，指定绘制点的尺寸。

举例：

MATLAB 提供了一个稀疏矩阵 west0479，调用如下函数来查看该稀疏矩阵的结构图：

```
load west0479
spy(west0479)
```

结构图如图 10-1 所示。

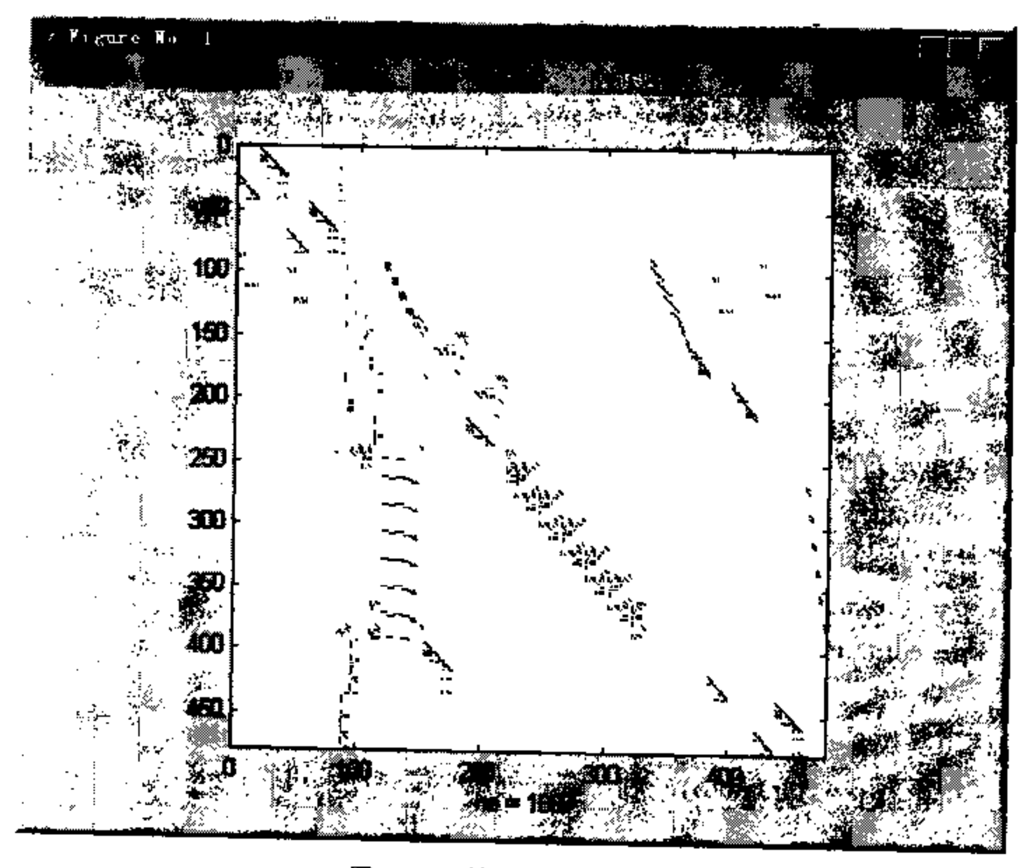


图 10-1　稀疏矩阵的结构图

若调用函数 spy(west0479,20)，则采用 20 个相素单位的点来绘制稀疏矩阵的结构图，结果如图 10-2 所示。

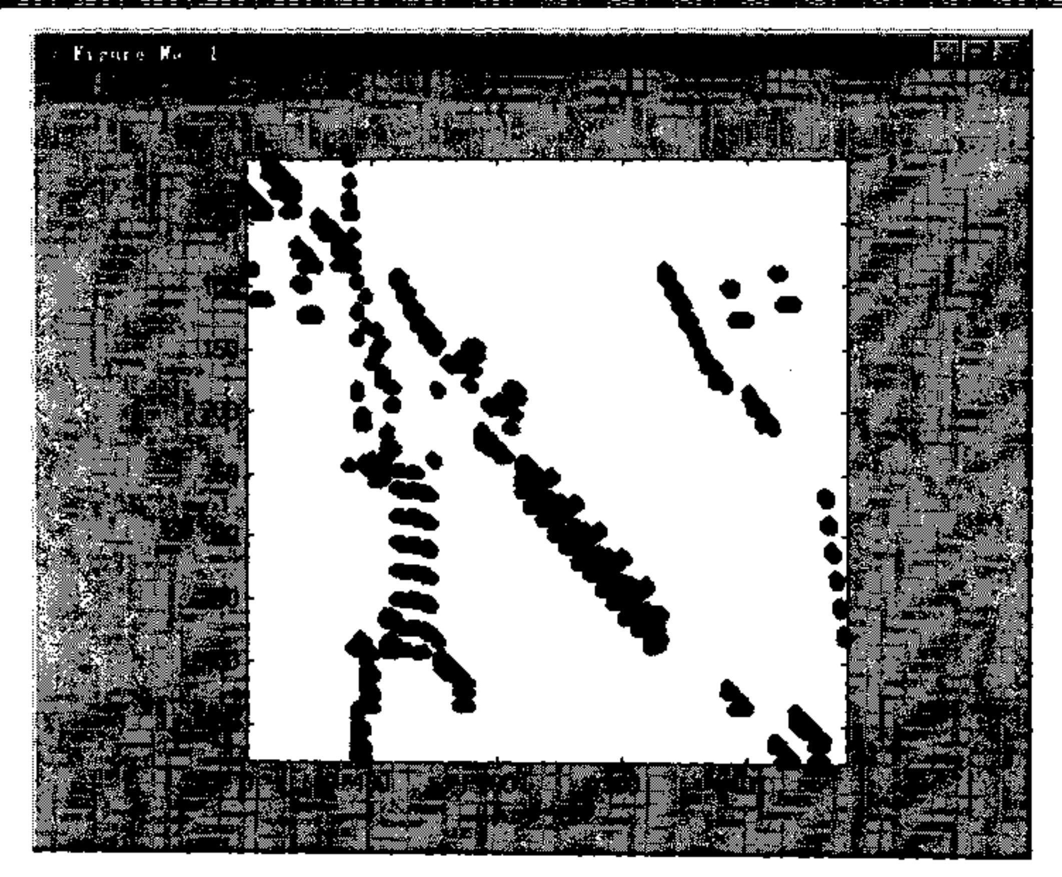


图 10-2 采用 20 个相素单位的点绘制稀疏矩阵的结构图

10.5 排序算法

1、 列最小度排序

名称：colmmd

进行列最小度排序。

语法：p = colmmd(S)

描述：p = colmmd(S)返回一个稀疏矩阵 S 的列最小度排序向量 p。

举例：

首先调用函数 S = sparse(wilkinson(21))生成一个 21×21 阶稀疏矩阵。

然后调用函数 p = colmmd(S)，对 S 进行列最小度排序，可得：

```
p =
  Columns 1 through 12
    11    3    1    2    4    5    8    6    7    9    10    13
  Columns 13 through 21
    12    21    20    19    18    17    16    14    15
```

为了比较起见，分别调用函数 spy(S)和 spy(S(:,p))将排序前和排序后的矩阵结构图绘制出来，如图 10-3 和 10-4 所示。

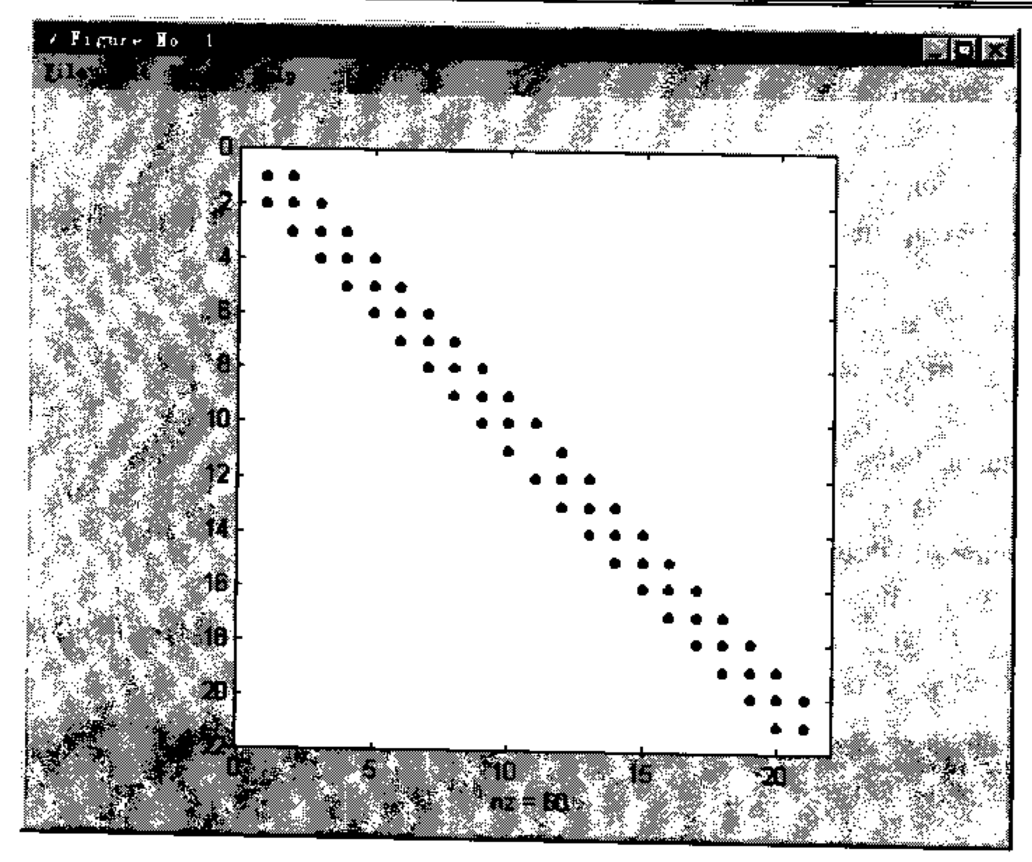


图 10-3　排序前的矩阵结构图

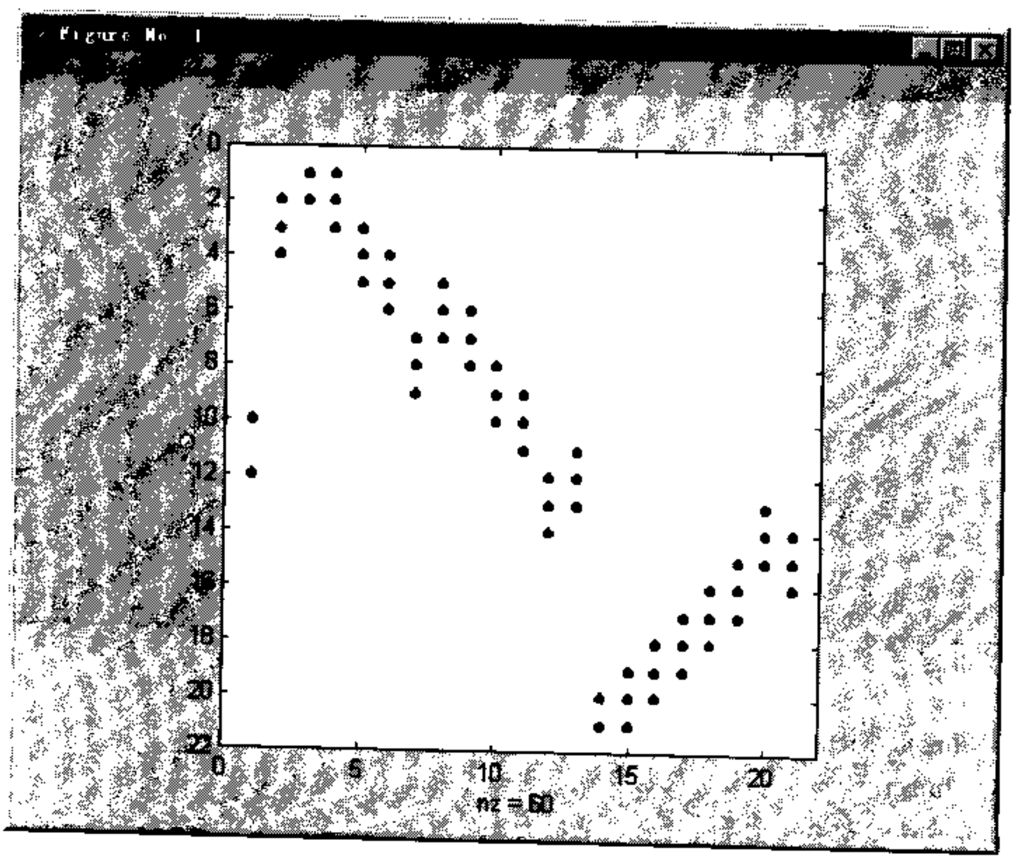


图 10-4　采用列最小度排序后的矩阵结构图

2. 非 0 算法列排序

名称：colperm

基于非 0 算法列排序。

语法：j = colperm(S)

描述：j = colperm(S)返回一个置换向量 j，并满足列向量 S(:,j)按非 0 元素升序排列。

举例：

首先调用函数 S = sparse(wilkinson(21))生成一个 21×21 阶稀疏矩阵。

然后调用函数 j = colperm(S)，对 S 进行基于非 0 算法的列排序，可得：

```
j =
  Columns 1 through 12
     1    11    21     2     3     4     5     6     7     8     9    10
  Columns 13 through 21
    12    13    14    15    16    17    18    19    20
```

为了比较起见，分别调用函数 spy(S)和 spy(S(:,j))，将排序前和排序后的矩阵结构图绘制出来，如图 10-3 和 10-5 所示。

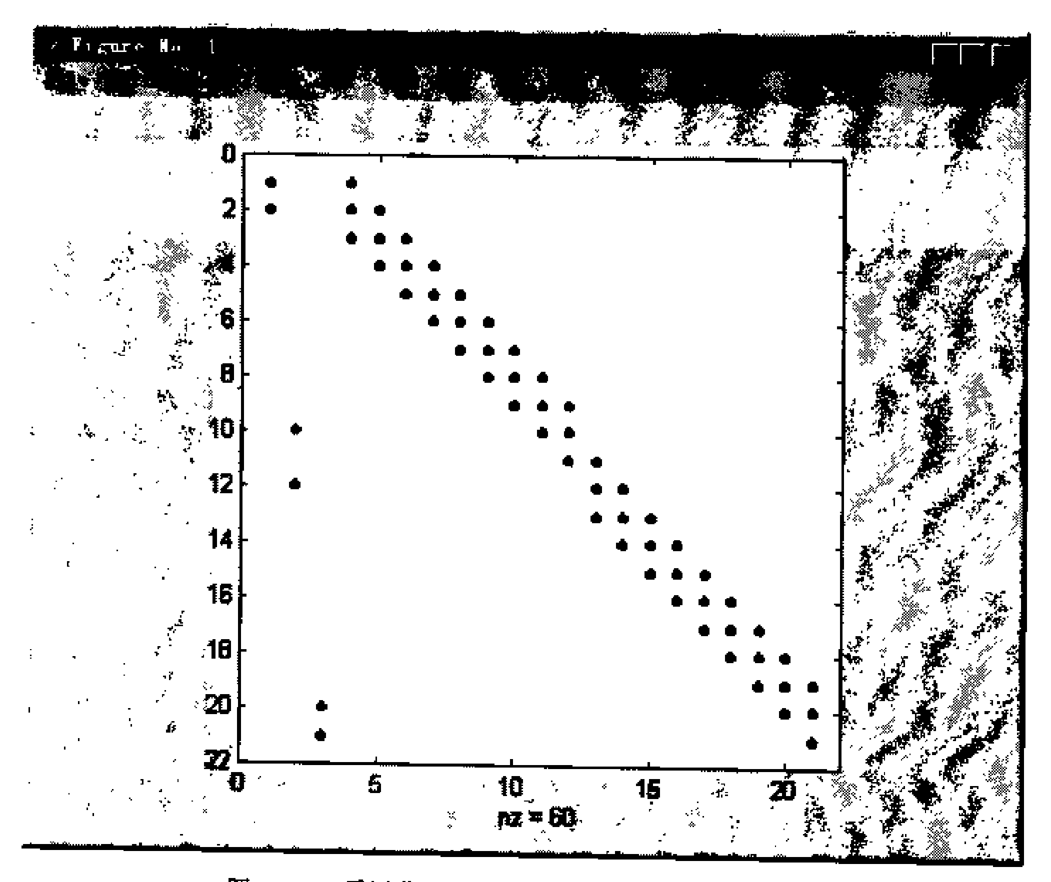


图 10-5　采用非 0 算法列排序后得到的矩阵结构图

3. Dulmage-Mendelsohn 分解

名称：dmperm

Dulmage-Mendelsohn 分解。

语法：该函数有如下几种表达形式：

- p = dmperm(A)
- [p,q,r] = dmperm(A)
- [p,q,r,s] = dmperm(A)

描述：p = dmperm(A)返回一个行置换向量 p，而且当 A 的行向量为满秩时，A(p,:)是一个包含非 0 对角元素的方阵。

当 A 是一个方阵时，[p,q,r] = dmperm(A)返回一个行置换向量 p 和列置换向量 q，并满足 A(p,q)为上三角矩阵。r 是一个描述三角形块边界的整数向量。

当 A 不是方阵时，[p,q,r,s] = dmperm(A)返回一个行置换向量 p、列置换向量 q 和索引向量 r 和 s，并满足 A(p,q)为上三角矩阵形式的条件。

举例：

首先调用函数 A = sparse(wilkinson(21))生成一个 21×21 阶稀疏矩阵。

然后调用函数 p = dmperm(A)，得到一个如下所示的行置换向量：

```
p =
  Columns 1 through 12
     1     2     3     4     5     6     7     8     9    10    12    11
  Columns 13 through 21
    13    14    15    16    17    18    19    20    21
```

若调用函数[p,q,r] = dmperm(A)，则可得到：

```
p =
  Columns 1 through 12
    21    20    19    18    17    16    15    14    13    12    11    10
  Columns 13 through 21
     9     8     7     6     5     4     3     2     1
q =
  Columns 1 through 12
    21    20    19    18    17    16    15    14    13    11    12    10
  Columns 13 through 21
     9     8     7     6     5     4     3     2     1
r =
     1    22
```

4. 随机置换

名称：randperm

随机置换。

语法：p = randperm(n)

描述：p = randperm(n)返回一个随机置换向量。

举例：

例如调用函数 p = randperm(6)时，可以得到如下所示的随机置换向量：

```
p =
     4     2     5     3     6     1。
```

5. 对称最小度排序

名称：symmmd

对称最小度排序。

语法：p = symmmd(S)

描述：p = symmmd(S)返回矩阵 S 的对称最小度排序。

举例：

首先调用函数 B = bucky+4*speye(60)得到一个 60×60 阶稀疏矩阵。

然后调用如下两个函数对矩阵 B 进行排序：

```
r = symrcm(B)
p = symmmd(B)
```

其中，对称最小度排序的结果为：

```
p =
  Columns 1 through 12
    51    23    48    24    52     3    21     4     1     7     6    26
  Columns 13 through 24
    28    27    30     5    25     9    39    38    50    42    44    43
  Columns 25 through 36
    46    56    58    57    60    59    45    41    31    54    17    19
  Columns 37 through 48
    18    20    11    13    12    33    36    34    14    15    32    40
  Columns 49 through 60
    35    37    10    55    49     8    29    47    53     2    22    16
```

调用如下两个函数，得到两个矩阵 B 重新排序后的矩阵。

```
R = B(r,r)
S = B(p,p)
```

为了直观起见，调用如下所示的函数，在窗口中将这些矩阵的结构图显示出来：

```
subplot(2,2,1)
spy(R)
title('B(r,r)')
subplot(2,2,2)
spy(S)
title('B(s,s)')
subplot(2,2,3)
spy(chol(R))
title('chol(B(r,r))')
subplot(2,2,4)
spy(chol(S))
title('chol(B(s,s))')
```

绘制结果如图 10-6 所示。

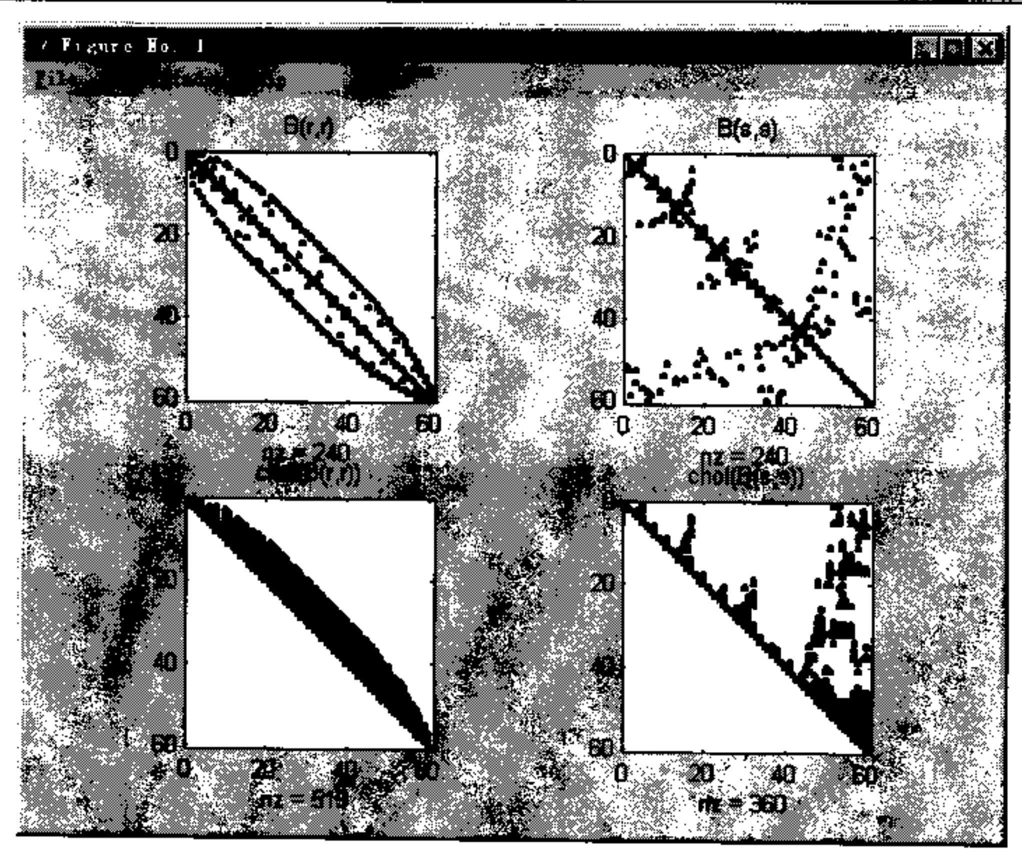


图 10-6 经过重排序后的矩阵结构图

6. 反向 Cuthill-McKee 排序

名称：symrcm

反向 Cuthill-McKee 排序。

语法：r = symrcm(S)

描述：r = symrcm(S)返回矩阵 S 的对称反向 Cuthill-McKee 排序。S(r,r)中的非 0 元素分布在对角线附近。

举例：

首先调用函数 B = bucky 得到一个 60×60 阶稀疏矩阵。

分别调用如下所示的三个函数，将该稀疏矩阵的结构图绘制出来(如图 7 所示)：

```
subplot(1,2,1)
spy(B)
title('B')
```

调用函数 p = symrcm(B)，得到矩阵 B 的反向 Cuthill-McKee 排序向量，结果为：

```
p =
  Columns 1 through 12
     1     6     2     5    10    11    12     7    26    30     3     4
  Columns 13 through 24
     9    15    13     8    27    29    16    17    21    25    38    14
  Columns 25 through 36
    37    42    28    43    20    18    22    24    39    33    36    41
```

```
Columns 37 through 48
  47    44    19    32    23    48    40    34    45    46    53    31
Columns 49 through 60
  52    49    57    35    58    50    54    51    56    59    55    60
```

调用函数 R = B(p,p)得到重新排序后的矩阵。

最后调用如下所示的三个函数，将重新排序后的矩阵的结构图显示出来(如图 10-7)：

```
subplot(1,2,2)
spy(R)
title('B(p,p)')
```

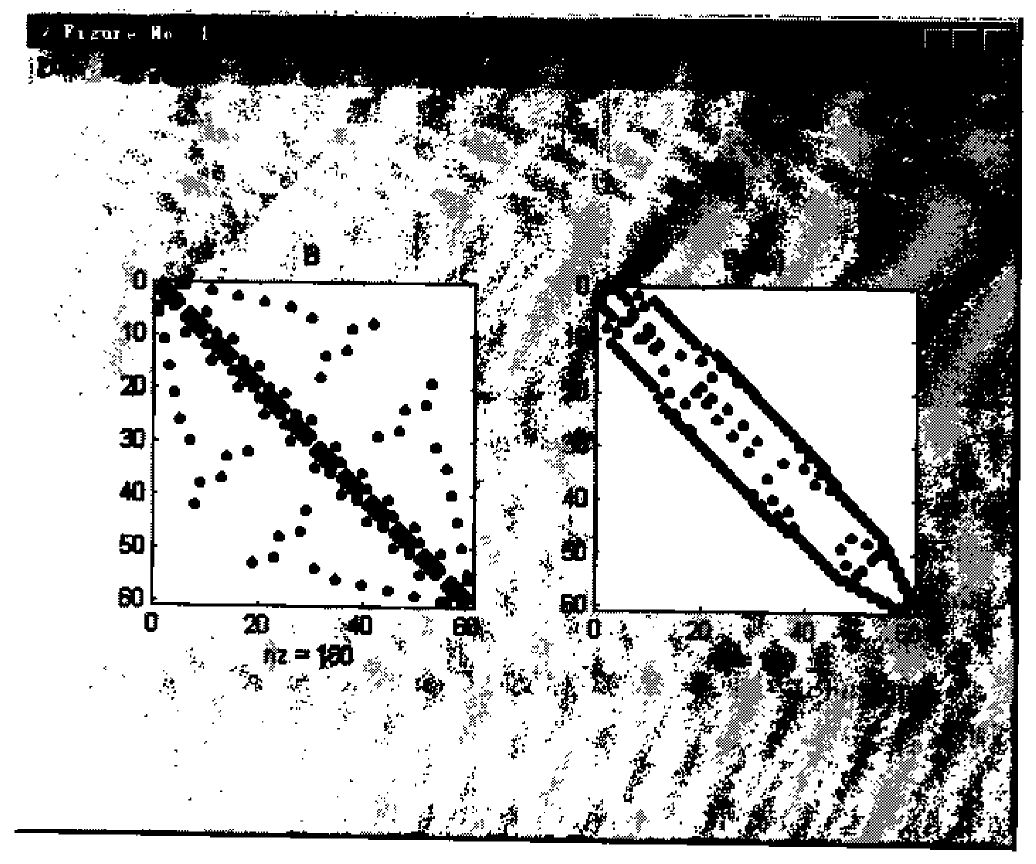


图 10-7　稀疏矩阵结构图比较

10.6　范数、条件数和秩

1．1 范数矩阵条件数

名称：condest

1 范数矩阵条件数的估计。

语法：该函数有如下两种表达形式：

- c = condest(A)
- [c,v] = condest(A)

描述：c = condest(A)使用 Hager 方法的 Higham 修正公式对矩阵的条件数进行估计。其中 c 是矩阵 A 为 1 阶范数的条件数。

[c,v] = condest(A)估计矩阵 A 的条件数，并返回一个向量 v，满足 $\|Av\| = \|A\|\|v\|/c$。

函数 condest 既可以处理实数矩阵，也可以处理复数矩阵。

举例：

首先调用函数 B = bucky 得到一个 60×60 阶稀疏矩阵。

然后调用函数 c = condest(A)，可以得到矩阵 A 的条件数为：

```
c =
    36.0000
```

2. 2 范数

名称：normest

2 范数估计。

语法：该函数有如下几种表达形式：

- nrm = normest(S)
- nrm = normest(S,tol)
- [nrm,count] = normest(...)

描述：函数 normest 最初是专门为稀疏矩阵设计的。但在 MATLAB 的最新版本中，该函数也可以用来处理满阵。

nrm = normest(S)返回矩阵 S 的 2 范数估计。

nrm = normest(S,tol)返回矩阵 S 的 2 范数估计。其中相对误差限使用 tol 指定的值，而不用其缺省值 1.e-6。

[nrm,count] = normest(...)返回矩阵的 2 范数估计。并给出已完成的幂迭代次数。

举例：

首先调用函数 W = gallery('wilkinson',101)，生成一个 101×101 阶 3 对角矩阵。

然后调用函数 nrm = normest(W)，估计 W 矩阵的 2 范数条件，结果为：

```
nrm =
    50.7458
```

10.7　线性方程的稀疏系统

在 MATLAB 中，系数矩阵为稀疏矩阵的线性方程组的解法可以分为两类：直接法和迭代法。一般来说，直接法都是高斯消去法的变种，直接法需对矩阵的每个元素进行操作。迭代法通过有限步迭代，得出方程组的近似解。一般情况下直接法比迭代法更快、使用范围更广，但直接法需要更多的内存空间。

当用直接法求解方程组时，一般不采用 lu、chol 等分解函数，而使用矩阵的左除和右除运算符。例如当矩阵 A 为方阵时，线性方程组 A*X=B 的解为 X=A\B；当矩阵 A 为长方阵时，采用最小二乘法求解。当矩阵 A 为方阵时，无论是满阵还是稀疏矩阵，A\B 占用的空间和 A 差不多，而计算方法将根据 A 的具体情况采用下面几种方法：

1，当 A 为三角阵时，对矩阵 B 的每一列直接计算。

2，当 A 为三角阵的初等变换时，先对 A 进行变换，再对矩阵 B 的每一列直接计算。

3，当 A 为对称或共轭对称且对角元素都为正时，采用 Cholesky 分解再对 B 的每列进行求解。

4，其他情况时，采用 LU 分解，再对矩阵 B 的每列进行求解。

MATLAB 提供了 6 个专门用于求解稀疏线性方程组的函数：bicg、bicgstab、cgs、gmres、pcg 和 qmr。这 6 个函数都用于解方程组 Ax=b，其中函数 pcg 只适用于对称正定矩阵 A，其他 5 个可解非对称方程。

1. 双共轭梯度法

名称：bicg

双共轭梯度法。

语法：该函数有如下几种表达形式：

- x = bicg(A,b)
- bicg(A,b,tol)
- bicg(A,b,tol,maxit)
- bicg(A,b,tol,maxit,M)
- bicg(A,b,tol,maxit,M1,M2)
- bicg(A,b,tol,maxit,M1,M2,x0)
- x = bicg(A,b,tol,maxit,M1,M2,x0)
- [x,flag] = bicg(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres] = bicg(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter] = bicg(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter,resvec] = bicg(A,b,tol,maxit,M1,M2,x0)

描述：函数 bicg 将从一个初始估计值(缺省时为一个长度为 n 的零向量)开始进行迭代，直到满足收敛限或达到最大允许迭代次数时，停止迭代。从数学上讲，就是当迭代值 x 所引起的相对残差 norm(b-A*x)/norm(b)小于或等于给定的收敛限时，停止迭代。该收敛限的缺省值为 1e-6。缺省的最大允许迭代次数为 n 和 20 二者之间的最小者。

x = bicg(A,b)返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。

当A不能够被显式地表达为矩阵时，用户可以将A表示为一个算子afun，其中函数afun(x)将返回矩阵与向量的内积 A*x，而函数 afun(x,'transp')将返回 A'*x。该算子既可以是一个 M 文件名，也可以是 MATLAB 的一个内置函数。在这种情况下，n 将被取为列向量 b 的长度。

bicg(A,b,tol)返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。

bicg(A,b,tol,maxit) 返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。参数 maxit 用来指定最大允许迭代次数。

bicg(A,b,tol,maxit,M)和 bicg(A,b,tol,maxit,M1,M2)通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中 A 必须为 n×n 阶方阵，而

且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。参数 maxit 用来指定最大允许迭代次数。

bicg(A,b,tol,maxit,M1,M2,x0) 通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。参数 maxit 用来指定最大允许迭代次数。参数 x0 用来指定初始估计值。若给定一个空矩阵[ ]，则缺省的初始估计值中全部为零向量。

x = bicg(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x。如果函数 bicg 迭代收敛，则在屏幕上会出现一个相应的信息。如果函数 bicg 迭代不收敛或已经达到了最大允许迭代次数或因意外情况程序被终止，此时会在屏幕上显示出一个警告信息，并给出迭代结束或终止时的相对残差 norm(b-A*x)/norm(b)和已经完成了的迭代次数。

[x,flag] = bicg(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 bicg 收敛性的参量 flag。该参量的取值和意义如表 10-1 中所示。

表 10-1 参量 flag 的取值和意义

Flag	意义
0	Bicg 函数正常收敛。
1	Bicg 函数的迭代次数已经达到了最大允许迭代次数，但还没有收敛。
2	形如 M*y = r 的方程组中的某个方程是病态的，不能够采用‘\’运算符计算得到一个可用解。
3	迭代非正常终止(两个连续的迭代步得到了完全相同的结果)。
4	函数 bicg 迭代中得到的某一个标量太大或太小，从而使得计算终止。

[x,flag,relres] = bicg(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 bicg 收敛性的参量 flag。该参量的取值和意义如表 10-1 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)。如果 flag = 0，则 relres≤tol。

[x,flag,relres,iter] = bicg(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 bicg 收敛性的参量 flag。该参量的取值和意义如表 1 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter，而且满足 0≤iter≤maxit。

[x,flag,relres,iter,resvec] = bicg(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 bicg 收敛性的参量 flag。该参量的取值和意义如表 1 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter 以及在每一迭代步中得到的残差范数。

举例：

调用如下所示的函数得到矩阵 A 和向量 b：

load west0479

A = west0479

b = sum(A,2)

现在用函数 bicg 求解方程组 A*x = b，在 MATLAB 命令窗口中输入如下命令：

[x,flag,relres,iter,resvec] = bicg(A,b)

结果为：

flag

```
    = 1
relres
    = 1
iter
    = 0,
resvec =
  1.0e+010 *
    0.0001
    0.0096
    0.0002
    0.0128
    0.0001
    4.9586
    4.5797
    0.0270
    0.0143
    0.0149
    0.0667
    0.0906
    0.1280
    0.1763
    0.1283
    0.0175
    0.0164
    0.0138
    0.0153
    0.0263
    0.0793
```

此时参数 flag 的取值为 1，表明函数 bicg 迭代了缺省的 20 步之后还没有收敛，因此自动终止计算。参数 iter 的取值为 0，表明初始估计的零向量要比随后迭代步中得到的任一向量都要好，产生这一问题的原因之一是选取的迭代方法性能太差。参数 relres 满足 relres=norm(b-A*x)/norm(b) = norm(b)/norm(b) = 1。

调用函数 semilogy(0:20,resvec/norm(b),'-o')，将迭代过程中各步求得的解显示出来，如图 10-8 所示。从图中可以看出，所得到的解振荡幅度比较大。

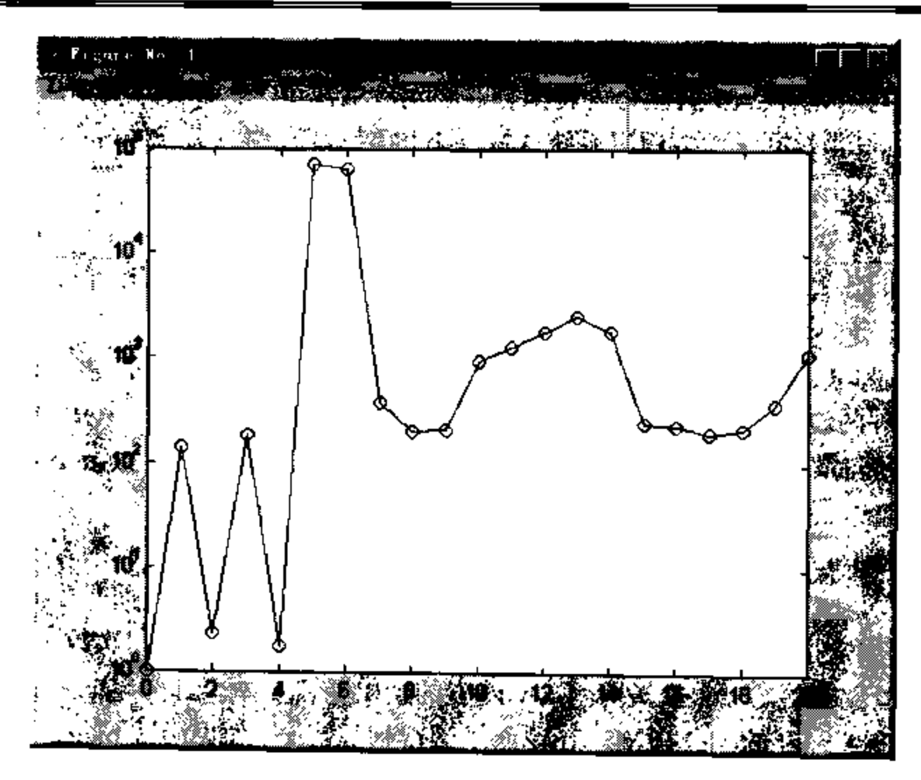


图 10-8　函数 bicg 得到的解

2. 双共轭梯度稳定法

名称：bicgstab

双共轭梯度稳定法。

语法：该函数有如下几种表达形式：

- x = bicgstab(A,b)
- bicgstab(A,b,tol)
- bicgstab(A,b,tol,maxit)
- bicgstab(A,b,tol,maxit,M)
- bicgstab(A,b,tol,maxit,M1,M2)
- bicgstab(A,b,tol,maxit,M1,M2,x0)
- x = bicgstab(A,b,tol,maxit,M1,M2,x0)
- [x,flag] = bicgstab(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres] = bicgstab(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter] = bicgstab(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter,resvec] = bicgstab(A,b,tol,maxit,M1,M2,x0)

描述：函数 bicgstab 将从一个初始估计值(缺省时为一个长度为 n 的零向量)开始进行迭代，直到满足收敛限或达到最大允许迭代次数时，停止迭代。从数学上讲，就是当迭代值 x 所引起的相对残差 norm(b-A*x)/norm(b)小于或等于给定的收敛限时，停止迭代。该收敛限的缺省值为 1e-6。缺省的最大允许迭代次数为 n 和 20 二者之间的最小者。

x = bicgstab(A,b)返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。

当 A 不能够被显式地表达为矩阵时，用户可以将 A 表示为一个算子 afun，其中函数 afun(x)

将返回矩阵与向量的内积 A*x，而函数 afun(x,'transp')将返回 A'*x。该算子既可以是一个 M 文件名，也可以是 MATLAB 的一个内置函数。在这种情况下，n 将被取为列向量 b 的长度。

bicgstab(A,b,tol)返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。

bicgstab(A,b,tol,maxit) 返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。参数 maxit 用来指定最大允许迭代次数。

bicgstab(A,b,tol,maxit,M) 和 bicgstab(A,b,tol,maxit,M1,M2) 通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。参数 maxit 用来指定最大允许迭代次数。

bicgstab(A,b,tol,maxit,M1,M2,x0) 通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。参数 maxit 用来指定最大允许迭代次数。参数 x0 用来指定初始估计值。若给定一个空矩阵[ ]，则缺省的初始估计值中全部为零向量。

x = bicgstab(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x。如果函数 bicgstab 迭代收敛，则在屏幕上会出现一个相应的信息。如果函数 bicgstab 迭代不收敛或已经达到了最大允许迭代次数或因意外情况程序被终止，此时会在屏幕上显示出一个警告信息，并给出迭代结束或终止时的相对残差 norm(b-A*x)/norm(b)和已经完成了的迭代次数。

[x,flag] = bicgstab(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 bicgstab 收敛性的参量 flag。该参量的取值和意义如表 1 中所示。

[x,flag,relres] = bicgstab(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 bicgstab 收敛性的参量 flag。该参量的取值和意义如表 1 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)。如果 flag = 0，则 relres≤tol。

[x,flag,relres,iter] = bicgstab(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 bicgstab 收敛性的参量 flag。该参量的取值和意义如表 1 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter，而且满足 0≤iter≤maxit。

[x,flag,relres,iter,resvec] = bicgstab(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 bicgstab 收敛性的参量 flag。该参量的取值和意义如表 1 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter 以及在每一迭代步中得到的残差范数。

举例：

调用如下所示的函数得到矩阵 A 和向量 b：

```
load west0479
A = west0479
b = sum(A,2)
```

现在用函数 bicgstab 求解方程组 A*x = b，在 MATLAB 命令窗口中输入如下命令：

```
[x,flag] = bicgstab(A,b)
```

结果为：

flag

= 1

参数 flag 的取值为 1，表明函数 bicgstab 经过缺省的 20 步迭代后还没有收敛到缺省的收敛限 1e-6。

若首先调用函数[L1,U1] = luinc(A,1e-5)对 A 矩阵进行分解，然后再调用函数[x1,flag1] = bicgstab(A,b,1e-6,20,L1,U1)对方程组 A*x = b 进行求解，则可得到：

flag1 =

2

参数 flag1 的取值为 2。这是由于上三角矩阵 U1 的某个对角线元素的值为 0，因此当函数 bicgstab 在第一步中采用'\'运算符求解方程 U1*y = r 时，该迭代步就被终止了。

现在提高 A 矩阵的分解精度，即调用函数[L2,U2] = luinc(A,1e-6)，然后再调用函数[x2,flag2,relres2,iter2,resvec2] = bicgstab(A,b,1e-15,10,L2,U2)对方程组 A*x = b 进行求解，则可得到：

```
flag2
      =   0
relres2
      = 3.4357e-016
iter2
      =6
resvec2 =
   1.0e+005 *
      7.0557
      0.0000
      0.0000
      0.0000
      0.0000
      0.0000
      0.0000
      0.0000
      0.0000
      0.0000
      0.0000
                                    0.0000
      0.0000
```

参数 flag2 的取值为 0，iter2 的取值为 6，这表明函数 bicgstab 在第 6 个迭代步收敛。

调用函数 semilogy(0:0.5:iter2,resvec2/norm(b),'-o')，将迭代过程中各步求得的解显示出来，如图 10-9 所示。

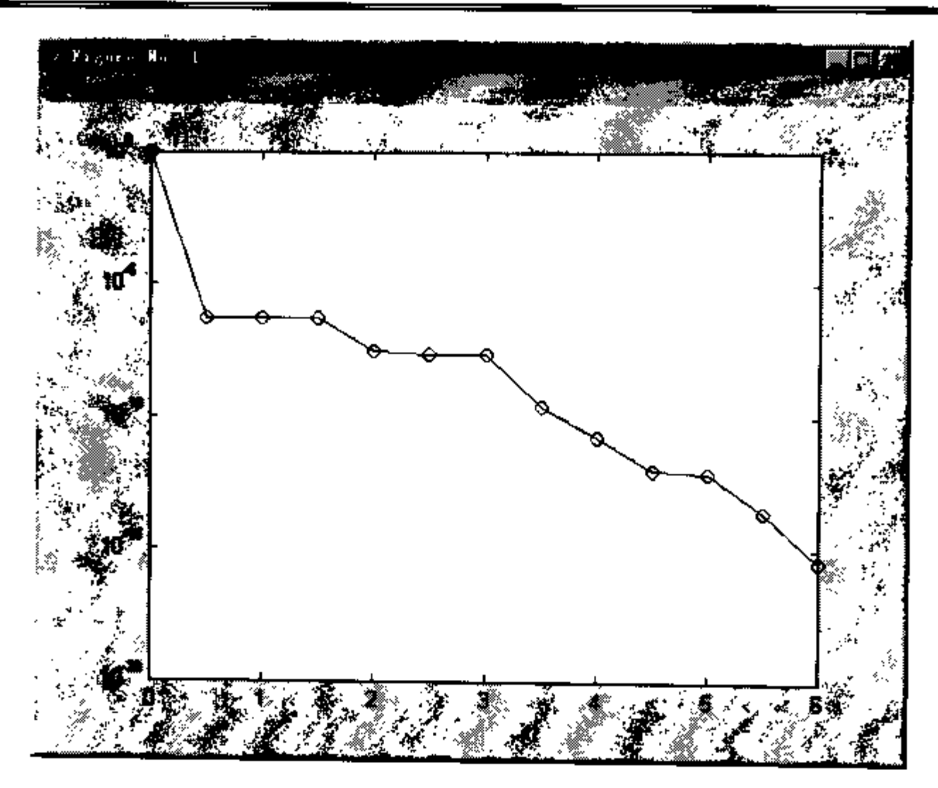


图 10-9 函数 bicgstab 得到的解

3. 二次共轭梯度法

名称：cgs

二次共轭梯度法。

语法：该函数有如下几种表达形式：

- x = cgs(A,b)
- cgs(A,b,tol)
- cgs(A,b,tol,maxit)
- cgs(A,b,tol,maxit,M)
- cgs(A,b,tol,maxit,M1,M2)
- cgs(A,b,tol,maxit,M1,M2,x0)
- x = cgs(A,b,tol,maxit,M1,M2,x0)
- [x,flag] = cgs(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres] = cgs(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter] = cgs(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter,resvec] = cgs(A,b,tol,maxit,M1,M2,x0)

描述：函数 cgs 将从一个初始估计值(缺省时为一个长度为 n 的零向量)开始进行迭代，直到满足收敛限或达到最大允许迭代次数时，停止迭代。从数学上讲，就是当迭代值 x 所引起的相对残差 norm(b-A*x)/norm(b)小于或等于给定的收敛限时，停止迭代。该收敛限的缺省值为 1e-6。缺省的最大允许迭代次数为 n 和 20 二者之间的最小者。

x = cgs(A,b)返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。

当A不能够被显式地表达为矩阵时，用户可以将A表示为一个算子 afun，其中函数 afun(x)

将返回矩阵与向量的内积 A*x，而函数 afun(x,'transp')将返回 A'*x。该算子既可以是一个 M 文件名，也可以是 MATLAB 的一个内置函数。在这种情况下，n 将被取为列向量 b 的长度。

cgs(A,b,tol)返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。

cgs(A,b,tol,maxit) 返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。参数 maxit 用来指定最大允许迭代次数。

cgs(A,b,tol,maxit,M)和 cgs(A,b,tol,maxit,M1,M2)通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。参数 maxit 用来指定最大允许迭代次数。

cgs(A,b,tol,maxit,M1,M2,x0) 通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n。参数 tol 用来指定迭代收敛限。参数 maxit 用来指定最大允许迭代次数。参数 x0 用来指定初始估计值。若给定一个空矩阵[ ]，则缺省的初始估计值中全部为零向量。

x = cgs(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x。如果函数 cgs 迭代收敛，则在屏幕上会出现一个相应的信息。如果函数 cgs 迭代不收敛或已经达到了最大允许迭代次数或因以外情况程序被终止，此时会在屏幕上显示出一个警告信息，并给出迭代结束或终止时的相对残差 norm(b-A*x)/norm(b)和已经完成了的迭代次数。

[x,flag] = cgs(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 cgs 收敛性的参量 flag。该参量的取值和意义如表 1 中所示。

[x,flag,relres] = cgs(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 cgs 收敛性的参量 flag。该参量的取值和意义如表 1 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)。如果 flag = 0，则 relres≤tol。

[x,flag,relres,iter] = cgs(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 cgs 收敛性的参量 flag。该参量的取值和意义如表 1 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter，而且满足 0≤iter≤maxit。

[x,flag,relres,iter,resvec] = cgs(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 cgs 收敛性的参量 flag。该参量的取值和意义如表 1 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter 以及在每一迭代步中得到的残差范数。

举例：

调用如下所示的函数得到矩阵 A 和向量 b：

```
load west0479
A = west0479
b = sum(A,2)
```

现在用函数 cgs 求解方程组 A*x = b，在 MATLAB 命令窗口中输入如下命令：

```
[x,flag] = cgs(A,b)
```

结果为：

flag =

  1

此时参数 flag 的取值为 1，表明函数 cgs 迭代了缺省的 20 步之后还没有收敛，因此自动终止计算。

若首先调用函数[L1,U1] = luinc(A,1e-5)对 A 矩阵进行分解，然后再调用函数[x1,flag1] = cgs(A,b,1e-6,20,L1,U1)对方程组 A*x = b 进行求解，则可得到：

flag1 =

  2

参数 flag1 的取值为 2。这是由于上三角矩阵 U1 的某个对角线元素的值为 0，因此当函数 bicgstab 在第一步中采用 '\' 运算符求解方程 U1*y = r 时，该迭代步就被终止了。

现在提高 A 矩阵的分解精度，即调用函数[L2,U2] = luinc(A,1e-6)，然后再调用函数[x2,flag2,relres2,iter2,resvec2] = cgs(A,b,1e-15,10,L2,U2)对方程组 A*x = b 进行求解，则可得：

```
flag2 = 0
relres2 =8.2702e-016
iter2 = 5
resvec2 =
  1.0e+005 *
    7.0557
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
```

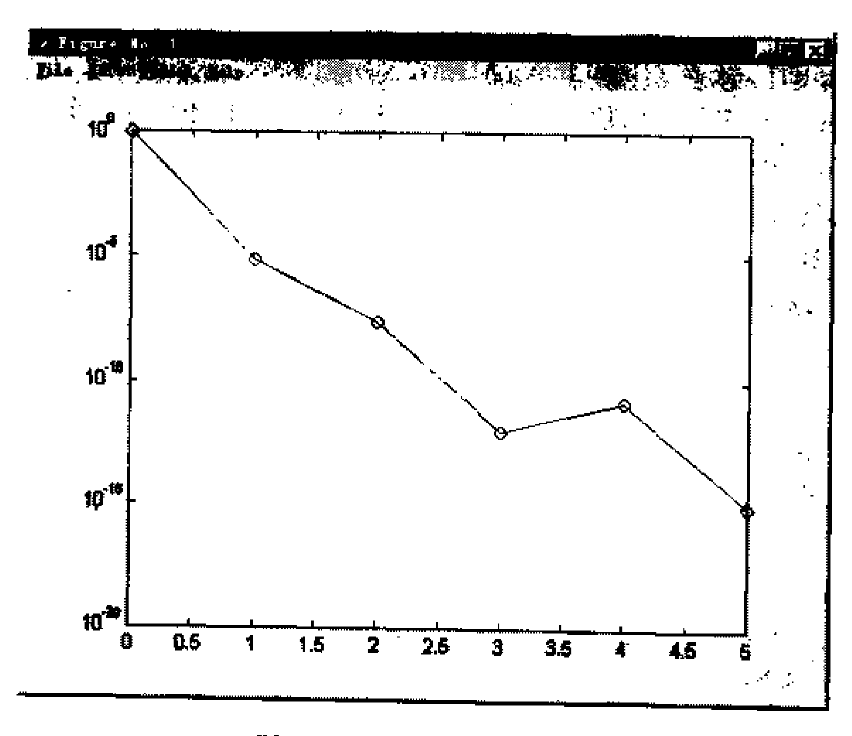


图 10-10 函数 cgs 得到的解

参数 flag2 的取值为 0，iter2 的取值为 5，这表明函数 bicgstab 在第 5 个迭代步正常收敛。

调用函数 semilogy(0:iter2, resvec2/norm(b),'-o')，将迭代过程中各步求得的解显示出来，如图 10-10 所示。

4．不完全 Cholesky 分解

名称：cholinc

稀疏矩阵的不完全 Cholesky 分解。

语法：该函数有如下几种表达形式：

- R = cholinc(X,droptol)
- R = cholinc(X,options)
- R = cholinc(X,'0')
- [R,p] = cholinc(X,'0')
- R = cholinc(X,'inf')

描述：稀疏矩阵作为矩阵的一种表达方式，在本质上和满阵没有什么不同。只是由于空间和计算时间的需要，才把满阵压缩成稀疏矩阵。和满阵一样，稀疏矩阵也可以进行 LU 分解、Cholesky 分解、正交分解，还有稀疏矩阵特有的不完全分解。由于稀疏矩阵较大，MATLAB 还提供了一些专门用于系数阵为稀疏矩阵的线性方程组的函数，如前面已经介绍过的 bicg、bicgstab 等。此处介绍对稀疏矩阵进行不完全 Cholesky 分解的函数 cholinc。其调用格式为：

R = cholinc(X,droptol)对矩阵 X 进行不完全 Cholesky 分解，其下降允许限为 droptol。

R = cholinc(X,options)对不完全 Cholesky 分解设置附加选项。其中选项结构 options 中可以包含如表 10-2 中所示的一些参量。

表 10-2　options 结构中可以包含的参量

参量名	意义
Droptol	不完全 Cholesky 分解的下降允许限。
Michol	修正的不完全 Cholesky 分解。
Rdiag	替换 R 矩阵对角线上的零元素。

其中参量 droptol 是一个非负数，当 droptol 为 0 时(缺省情况)，蜕化为完全 Cholesky 分解。参量 michol 的取值可以为 0 或 1，当取 0 时，表示不对不完全 Cholesky 分解进行修正；取 1 时(缺省情况)，表示对不完全 Cholesky 分解进行修正。参量 rdiag 的取值可以为 0 或 1，缺省为 0，当取 1 时，上三角矩阵 R 中的任何零对角元素将会被替换为局部下降允许限的平方根。

R = cholinc(X,'0')对一个对称正定的稀疏矩阵 X 进行不完全 Cholesky 分解，将得到的上三角分解因子矩阵返回到 R。矩阵 R 和 triu(X)具有相同的密度。X 的下三角分解因子矩阵为上三角分解因子矩阵的转置。

需要指出的是，矩阵 X 的正定性并不能保证一定存在如上所述的上三角分解因子矩阵。如果 X 不能进行不完全 Cholesky 分解，此时会在窗口中出现一个错误信息；如果 X 能够进行正常的不完全 Cholesky 分解，则矩阵 R'*R 和矩阵 X 在结构图上应该一致。

[R,p] = cholinc(X,'0')如果 R 存在，则 p = 0；如果 R 不存在，则 p 是一个正整数。

R = cholinc(X,'inf') 对一个对称正定的稀疏矩阵 X 进行不完全 Cholesky 分解，将得到的

上三角分解因子矩阵返回到 R。

举例:

利用函数 S = delsq(numgrid('C',15))得到一个对称正定矩阵 S。

为了比较经过完全 Cholesky 分解和不完全 Cholesky 分解后矩阵 S 的变化情况。首先利用函数 spy(S)将原始矩阵 S 的结构图在窗口中显示出来，如图 10-11 所示。

然后调用函数 C = chol(S)对矩阵 S 进行完全 Cholesky 分解，并执行函数 spy(C)，得到矩阵 S 经过完全 Cholesky 分解后的结构图，如图 10-12 所示。

调用函数 R0 = cholinc(S,'0')对矩阵 S 进行 0 级不完全 Cholesky 分解，并执行函数 spy(R0)，得到矩阵 S 经过 0 级不完全 Cholesky 分解后的结构图，如图 10-13 所示。

比较图 10-12 和 10-13 可以看出，经过完全 Cholesky 分解后，沿着矩阵 S 的对角线半带宽区域被非 0 元素填充。而经过 0 级不完全 Cholesky 分解后，并不发生填充现象。

执行命令 S2 = S，将得到一个与矩阵 S 完全相同的矩阵 S2，然后执行 S2 (101,101) = 0 将 S2 中的对角线元素(101,101)取为 0 值，从而使得 S2 变为一个奇异矩阵。最后调用函数 [R,p]=cholinc(S2,'0')，对 S2 矩阵进行 0 级不完全 Cholesky 分解。该不完全 Cholesky 分解在计算到矩阵 S2 的底 101 行时停止。

调用函数 spy(R),可以得到 S2 矩阵经过 0 级不完全 Cholesky 分解后的结构图,如图 10-14 所示。

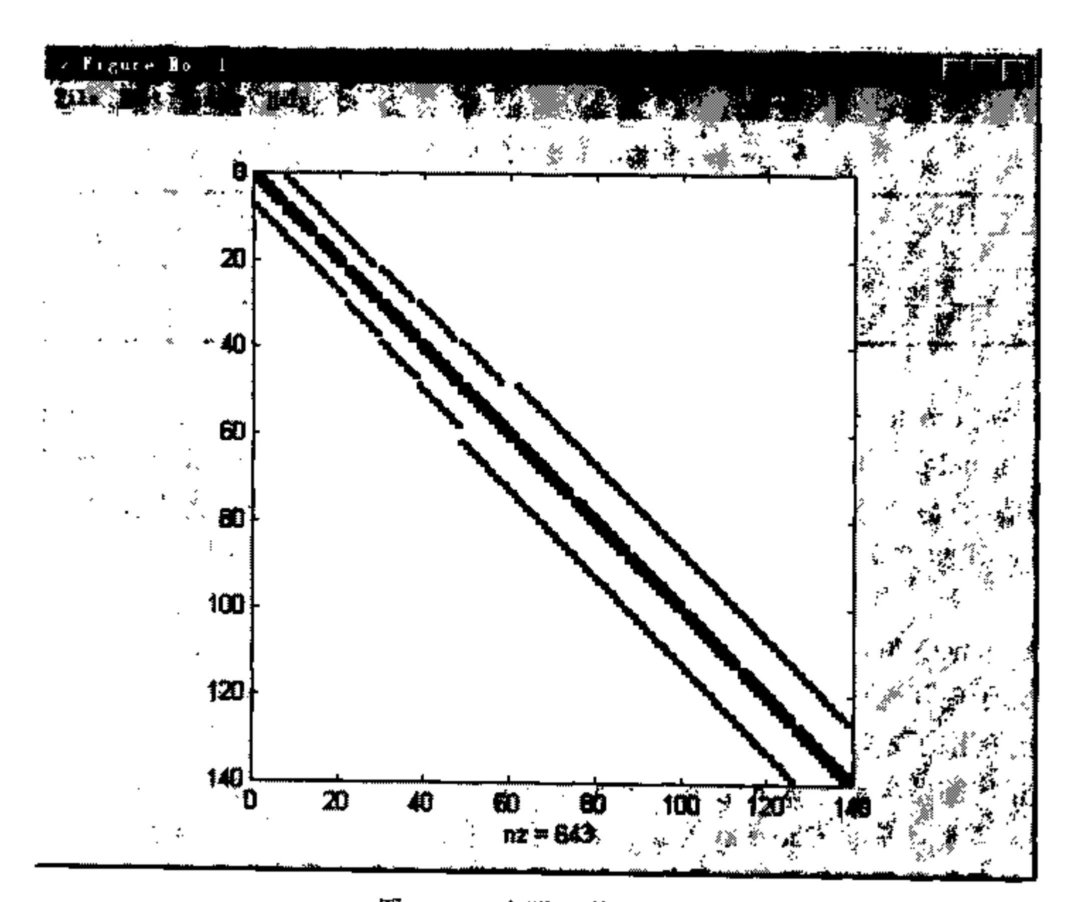


图 10-11　矩阵 S 的结构图

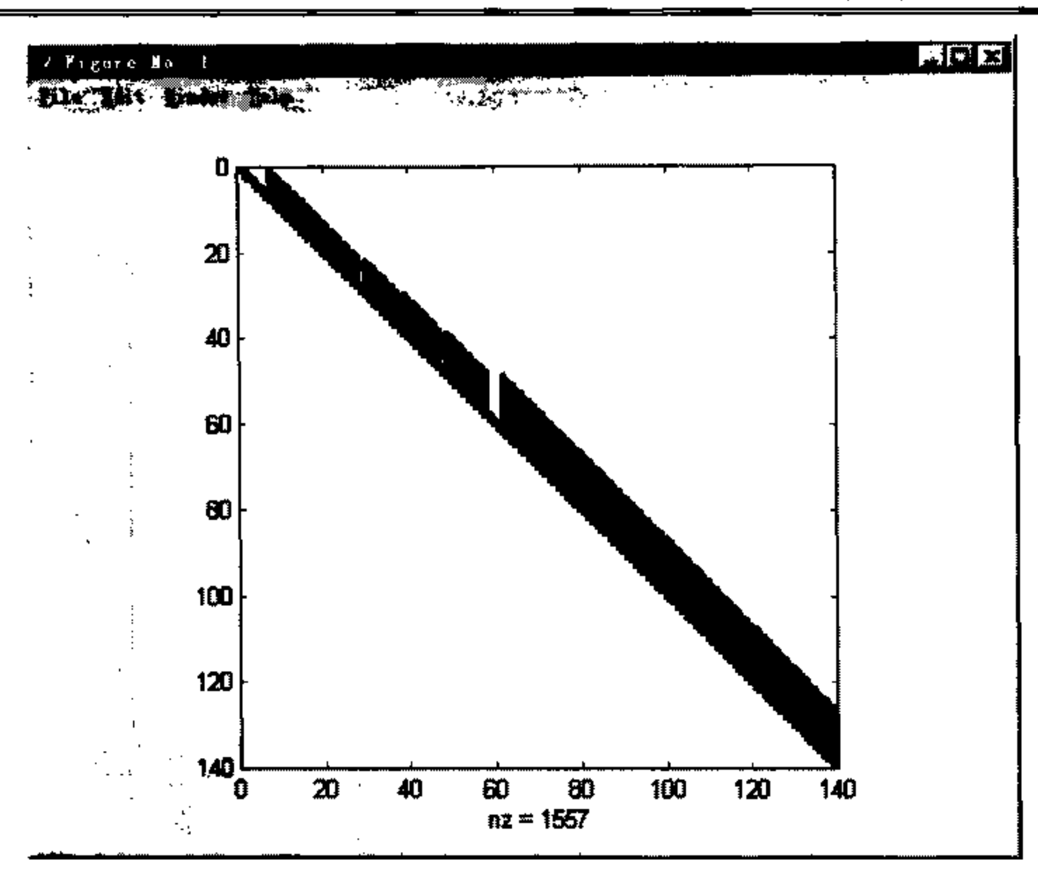


图 10-12　矩阵 S 经过完全 Cholesky 分解后的结构图

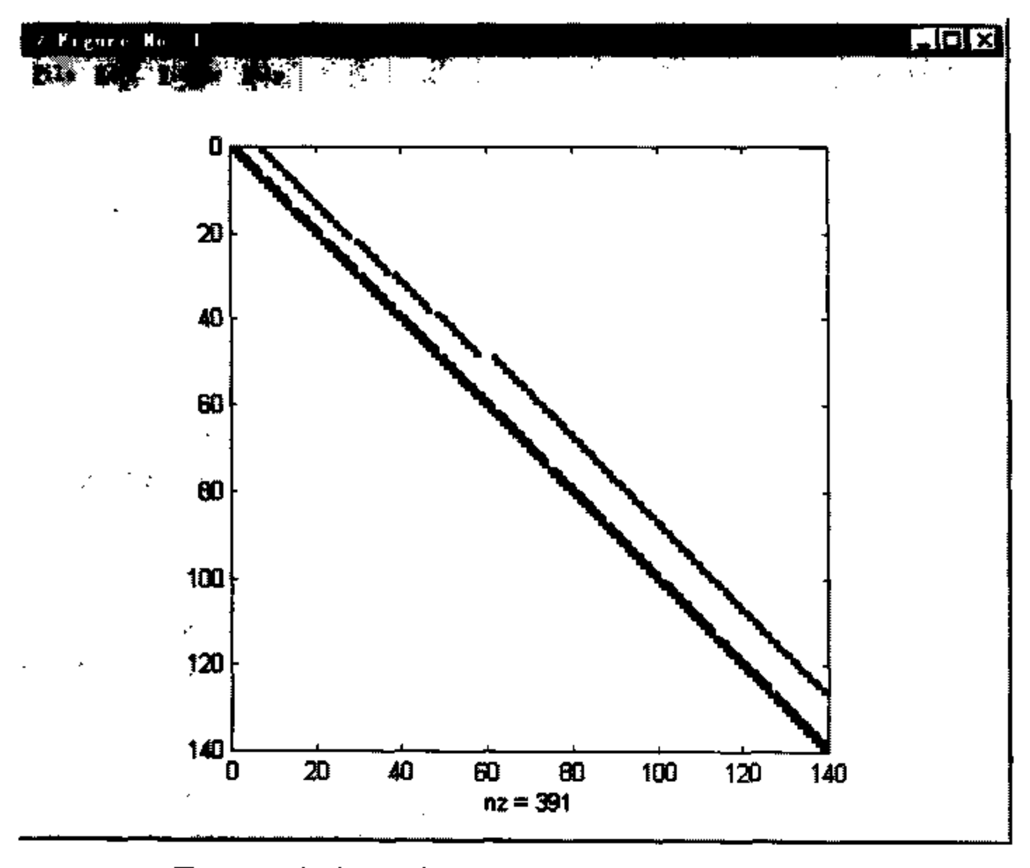


图 10-13　矩阵 S 经过 0 级不完全 Cholesky 分解后的结构图

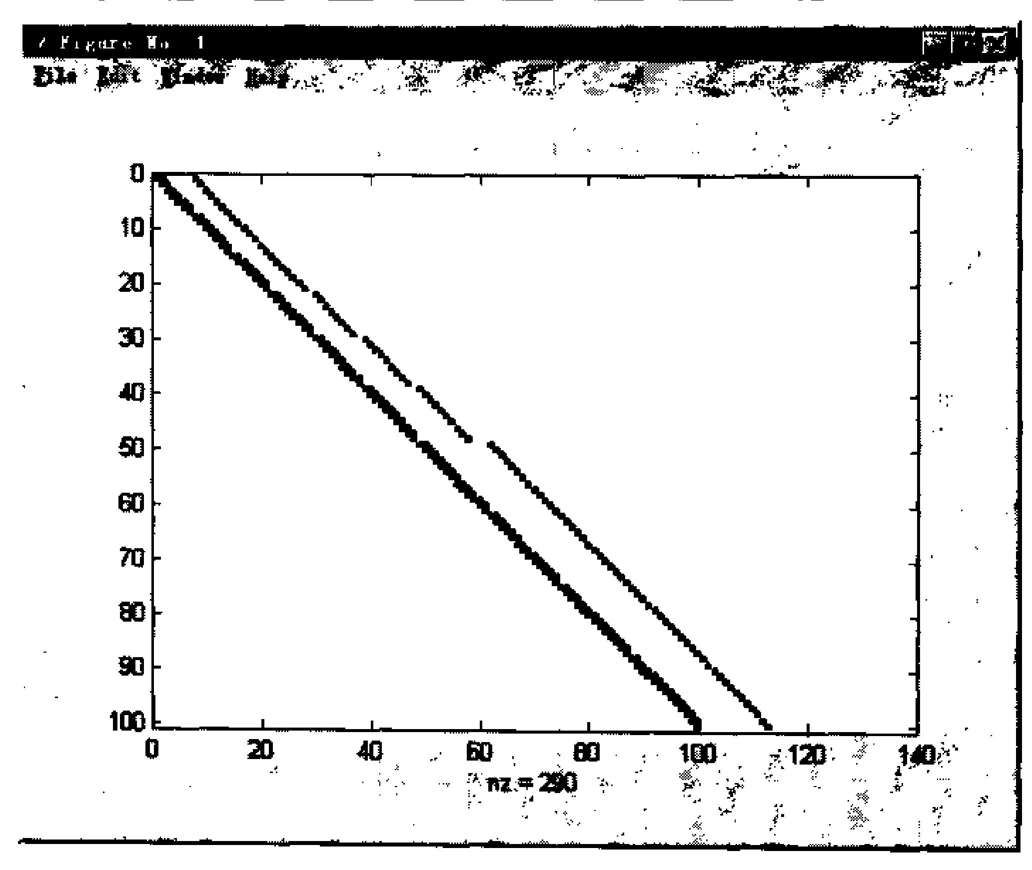


图 10-14 矩阵 S2 经过 0 级不完全 Cholesky 分解后的结构图

5. Cholesky 分解的秩 1 修正

名称：cholupdate

Cholesky 分解的秩 1 修正。

语法：该函数有如下几种表达形式：

- R1 = cholupdate(R,x)
- R1 = cholupdate(R,x,'+')
- R1 = cholupdate(R,x,'-')
- [R1,p] = cholupdate(R,x,'-')

描述：R1 = cholupdate(R,x)返回矩阵 A + x*x'的上三角 Cholesky 分解因子矩阵。其中 R = chol(A)是矩阵 A 的原始 Cholesky 分解因子矩阵，x 是相应长度的列向量。

需要指出的是，函数 cholupdate 只能处理 R 为上三角矩阵或对角矩阵的情形，若 R 为下三角矩阵，则函数不能正常执行。

R1 = cholupdate(R,x,'+')与函数 R1 = cholupdate(R,x)相同。

R1 = cholupdate(R,x,'-')返回矩阵 A - x*x'的上三角 Cholesky 分解因子矩阵。如果 R 不是一个正确的 Cholesky 分解因子矩阵或正定的下降矩阵，则会在窗口中输出一个错误信息。

[R1,p] = cholupdate(R,x,'-')不会输出错误信息，计算结果的性质通过参数 p 来体现，具体来讲：

如果 p = 0，则表示 R1 是矩阵 A - x*x'的上三角 Cholesky 分解因子矩阵；

如果 p > 0，则表示 R1 是原始矩阵 A 的上三角 Cholesky 分解因子矩阵；

如果 p = 1，则表示由于下降矩阵不正定，使得函数 cholupdate 运行失败；

如果 p = 2，则表示由于上三角矩阵 R 不是一个正确的 Cholesky 分解因子矩阵，使得函数 cholupdate 运行失败。

举例：

对于如下所示的 4×4 阶矩阵：

```
A =
     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
```

首先调用函数 R = chol(A)对 A 进行 Cholesky 分解，可得到一个如下所示的上三角矩阵：

```
R =
     1     1     1     1
     0     1     2     3
     0     0     1     3
     0     0     0     1
```

任选一个列向量

```
x =
     0
     0
     0
     1
```

调用函数 R1 = cholupdate(R,x)，返回如下所示的 A + x*x'的上三角 Cholesky 分解因子矩阵：

```
R1 =
    1.0000    1.0000    1.0000    1.0000
         0    1.0000    2.0000    3.0000
         0         0    1.0000    3.0000
         0         0         0    1.4142
```

此时若调用函数 R1 = cholupdate(R,x,'-')，会得到如下所示的结果：

```
??? Error using ==> cholupdate
Downdated matrix must be positive definite.
```

这表明，由于矩阵 A - x*x'不正定，因此函数 cholupdate 不能正常运行。

若把向量 x 改为：

```
x =
         0
         0
         0
    0.5000
```

此时矩阵 A - x*x'正定。因此当再调用函数 R1 = cholupdate(R,x,'-')时，会得到：

```
R1 =
    1.0000    1.0000    1.0000    1.0000
         0    1.0000    2.0000    3.0000
         0         0    1.0000    3.0000
         0         0         0    0.8660
```

6. 广义最小残差法

名称：gmres

广义最小残差法。

语法：该函数有如下几种表达形式：

- x = gmres(A,b,restart)
- gmres(A,b,restart,tol)
- gmres(A,b,restart,tol,maxit)
- gmres(A,b,restart,tol,maxit,M)
- gmres(A,b,restart,tol,maxit,M1,M2)
- gmres(A,b,restart,tol,maxit,M1,M2,x0)
- x = gmres(A,b,restart,tol,maxit,M1,M2,x0)
- [x,flag] = gmres(A,b,restart,tol,maxit,M1,M2,x0)
- [x,flag,relres] = gmres(A,b,restart,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter] = gmres(A,b,restart,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter,resvec] = gmres(A,b,restart,tol,maxit,M1,M2,x0)

描述：函数 gmres 将从一个初始估计值(缺省时为一个长度为 n 的零向量)开始进行迭代。每隔 restart 个迭代步，gmres 将进行重启动，而且以上一个迭代循环中得到的结果为新的循环迭代中的起始值。当得到的解满足收敛限或达到最大允许迭代次数时，停止迭代。从数学上讲，就是当迭代值 x 所引起的相对残差 norm(b-A*x)/norm(b)小于或等于给定的收敛限时，停止迭代。该收敛限的缺省值为 1e-6。缺省的最大允许迭代次数为 n/restart 和 10 二者之间的最小者。

x = gmres(A,b,restart) 返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n，参数 restart 用来指定进行重启动前的迭代次数。

当A不能够被显式地表达为矩阵时，用户可以将A表示为一个算子 afun，其中函数 afun(x)将返回矩阵与向量的内积 A*x，而函数 afun(x,'transp')将返回 A'*x。该算子既可以是一个 M 文件名，也可以是 MATLAB 的一个内置函数。在这种情况下，n 将被取为列向量 b 的长度。

gmres(A,b,restart,tol)返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n，参数 restart 用来指定进行重启动前的迭代次数，参数 tol 用来指定迭代收敛限。

gmres(A,b,restart,tol,maxit)返回线性方程组系统 A*x = b 的解 x。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n，参数 restart 用来指定进行重启动前的迭代次数，参数 tol 用来指定迭代收敛限，参数 maxit 用来指定最大允许迭代次数。

gmres(A,b,restart,tol,maxit,M) 和 gmres(A,b,restart,tol,maxit,M1,M2) 通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中

A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n，参数 restart 用来指定进行重启动前的迭代次数，参数 tol 用来指定迭代收敛限，参数 maxit 用来指定最大允许迭代次数。

gmres(A,b,restart,tol,maxit,M1,M2,x0)通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中 A 必须为 n×n 阶方阵，而且列向量 b 的长度为 n，参数 restart 用来指定进行重启动前的迭代次数，参数 tol 用来指定迭代收敛限，参数 maxit 用来指定最大允许迭代次数，参数 x0 用来指定初始估计值。若给定一个空矩阵[ ]，则缺省的初始估计值中全部为零向量。

x = gmres(A,b,restart,tol,maxit,M1,M2,x0)返回求解得到的解 x。如果函数 gmres 迭代收敛，则在屏幕上会出现一个相应的信息。如果函数 gmres 迭代不收敛或已经达到了最大允许迭代次数或因以外情况程序被终止，此时会在屏幕上显示出一个警告信息，并给出迭代结束或终止时的相对残差 norm(b-A*x)/norm(b)和已经完成了的迭代次数。

[x,flag] = gmres(A,b,restart,tol,maxit,M1,M2,x0) 返回求解得到的解 x 和一个描述函数 gmres 收敛性的参量 flag。该参量的取值和意义如表 10-3 中所示。

表 10-3 参量 flag 的取值和意义

Flag	意义
0	Gmres 函数正常收敛。
1	Gmres 函数的迭代次数已经达到了最大允许迭代次数，但还没有收敛。
2	形如 M*y = r 的方程组中的某个方程是病态的，不能够采用‘\’运算符计算得到一个可用解。
3	迭代非正常终止(两个连续的迭代步得到了完全相同的结果)。

[x,flag,relres] = gmres(A,b,restart,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 gmres 收敛性的参量 flag。该参量的取值和意义如表 3 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)。如果 flag = 0，则 relres≤tol。

[x,flag,relres,iter] = gmres(A,b,restart,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 gmres 收敛性的参量 flag。该参量的取值和意义如表 3 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter，而且满足 0≤iter≤maxit。

[x,flag,relres,iter,resvec] = gmres(A,b,restart,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 gmres 收敛性的参量 flag。该参量的取值和意义如表 3 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter 以及在每一迭代步中得到的残差范数。

举例：

调用如下所示的函数得到矩阵 A 和向量 b：

```
load west0479
A = west0479
b = sum(A,2)
```

现在用函数 gmres 求解方程组 A*x = b，在 MATLAB 命令窗口中输入如下命令：

```
[x,flag] = gmres(A,b,5)
```

结果为：

```
flag
    = 1
```

参数 flag 的取值为 1，表明函数 gmres 经过缺省的 10 步迭代后还没有收敛到缺省的收敛限 1e-6。

若首先调用函数[L1,U1] = luinc(A,1e-5)对 A 矩阵进行分解，然后再调用函数[x1,flag1] = gmres(A,b,5,1e-6,5,L1,U1)对方程组 A*x = b 进行求解，则可得到：

```
flag1 =
     2
```

参数 flag1 的取值为 2。这是由于上三角矩阵 U1 的某个对角线元素的值为 0，因此当函数 gmres 在第一步中采用 '\' 运算符求解方程 U1*y = r 时，该迭代步就被终止了。

现在提高 A 矩阵的分解精度，即调用函数[L2,U2] = luinc(A,1e-6)，并令 tol = 1e-15，然后再调用函数[x4,flag4,relres4,iter4,resvec4] = gmres(A,b,4,tol,5,L2,U2)对方程组 A*x = b 进行求解，则可得到：

```
flag4 =
     0
relres4 =
  8.3219e-017
iter4 =
     4     4
```

若调用函数[x6,flag6,relres6,iter6,resvec6] = gmres(A,b,6,tol,3,L2,U2)进行求解，则可得：

```
flag6 =
     0
relres6 =
  1.8472e-016
iter6 =
     2     4
```

若调用函数[x8,flag8,relres8,iter8,resvec8] = gmres(A,b,8,tol,3,L2,U2)进行求解，则可得：

```
flag8 =
     0
relres8 =
  1.1805e-016
iter8 =
     2     2
```

参数 flag4、flag6 和 flag8 的取值为 0，这表明函数 gmres 已经收敛。

7. 不完全 LU 分解

名称：luinc

不完全 LU 分解。

语法：该函数有如下几种表达形式：

- luinc(X,'0')
- [L,U] = luinc(X,'0')
- [L,U,P] = luinc(X,'0')

- luinc(X,droptol)
- luinc(X,options)
- [L,U] = luinc(X,options)
- [L,U] = luinc(X,droptol)
- [L,U,P] = luinc(X,options)
- [L,U,P] = luinc(X,droptol)

描述：LU 分解采用高斯消去法分解稀疏矩阵。利用函数 luinc 可以对稀疏矩阵进行不完全 LU 分解，并且可以得到一个单位下三角矩阵、一个上三角矩阵和一个置换矩阵，其调用格式为：

luinc(X,'0')对稀疏方阵 X 进行 0 级不完全 LU 分解。

[L,U] = luinc(X,'0')返回一个置换矩阵与一个单位下三角矩阵的乘积 L 和一个上三角矩阵 U。其中 X 为 n×n 阶方阵，并满足：nnz(L)+nnz(U) = nnz(X)+n。

[L,U,P] = luinc(X,'0')返回一个单位下三角矩阵 L、一个上三角矩阵 U 和一个置换矩阵 P。其中 X 为 n×n 阶方阵，并满足：spones(U) = spones(triu(P*X))。

luinc(X,droptol)对任意的稀疏矩阵 X 进行不完全 LU 分解，其下降允许限为 droptol。

luinc(X,options)对不完全 LU 分解设置附加选项。其中选项结构 options 中可以包含 droptol、milu、udiag 和 thresh 等参量。其中：

参量 droptol 是一个非负数，用来指定不完全 LU 分解的下降允许限。

参量 milu 的取值可以为 0 或 1，当取 0 时(缺省情况)，表示不对不完全 LU 分解进行修正；取 1 时，表示对不完全 LU 分解进行修正。

参量 udiag 的取值可以为 0 或 1，缺省为 0，当取 1 时，上三角矩阵 R 中的任何零对角元素将会被替换为局部下降允许限。

参量 thresh 的取值范围为[0 1]，用来指定转轴极限。

[L,U] = luinc(X,options)返回一个单位下三角矩阵的初等变换 L 和上三角矩阵 U。而且 L*U 近似等于 X。

[L,U] = luinc(X,droptol)采用下降允许限进行不完全 LU 分解。返回一个单位下三角矩阵的初等变换 L 和上三角矩阵 U。而且 L*U 近似等于 X。

[L,U,P] = luinc(X,options)返回一个单位下三角矩阵 L、一个上三角矩阵 U 和一个置换矩阵 P。并且 U 的非零元素满足：abs(U(i,j)) >= droptol*norm((X:,j))。而且 L*U 近似等于 P*X。

[L,U,P] = luinc(X,droptol)采用下降允许限进行不完全 LU 分解。返回一个单位下三角矩阵 L、一个上三角矩阵 U 和一个置换矩阵 P。并且 U 的非零元素满足：abs(U(i,j)) >= droptol*norm((X:,j))。而且 L*U 近似等于 P*X。

举例：

调用如下所示的函数得到矩阵 A 和向量 b：

load west0479

S = west0479

首先执行函数 LU = lu(S)对稀疏矩阵 S 进行完全 LU 分解，并调用函数 spy(LU)绘制出 S 矩阵经过完全 LU 分解后的结构图，如图 10-15 所示。

然后调用函数[L,U,P] = luinc(S,'0')，对稀疏矩阵 S 进行 0 级不完全 LU 分解。调用函数

spy(L)绘制出下三角矩阵 L 的结构图，如图 10-16 所示：调用函数 spy(U)绘制出上三角矩阵 U 的结构图，如图 10-17 所示。

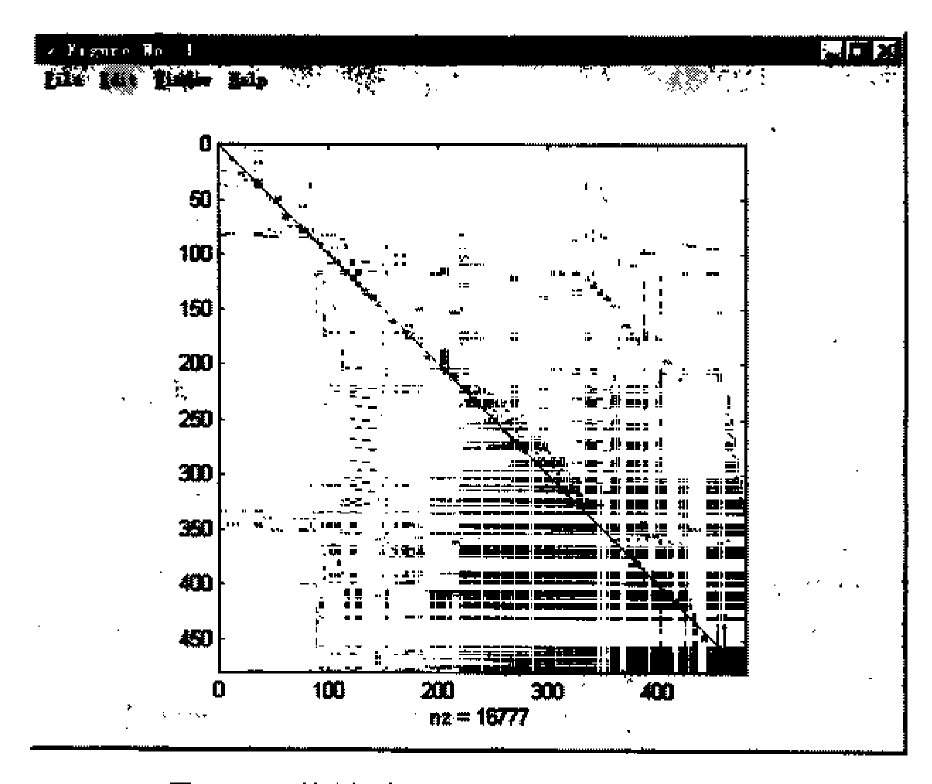


图 10-15 稀疏矩阵 S 经过完全 LU 分解后的结构图

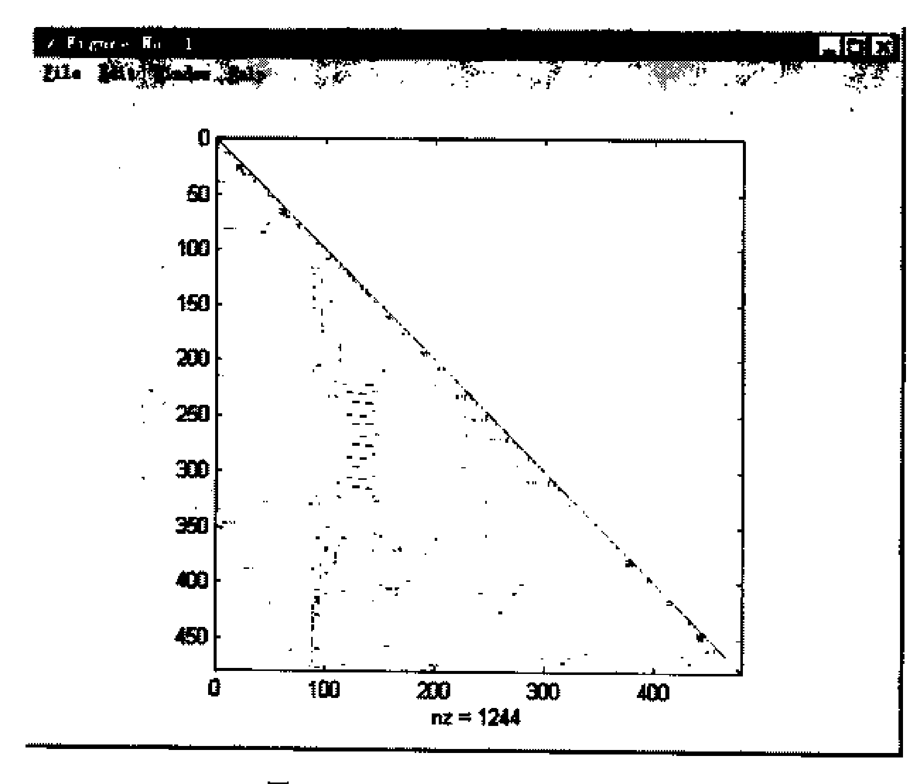


图 10-16 下三角矩阵 L 的结构图

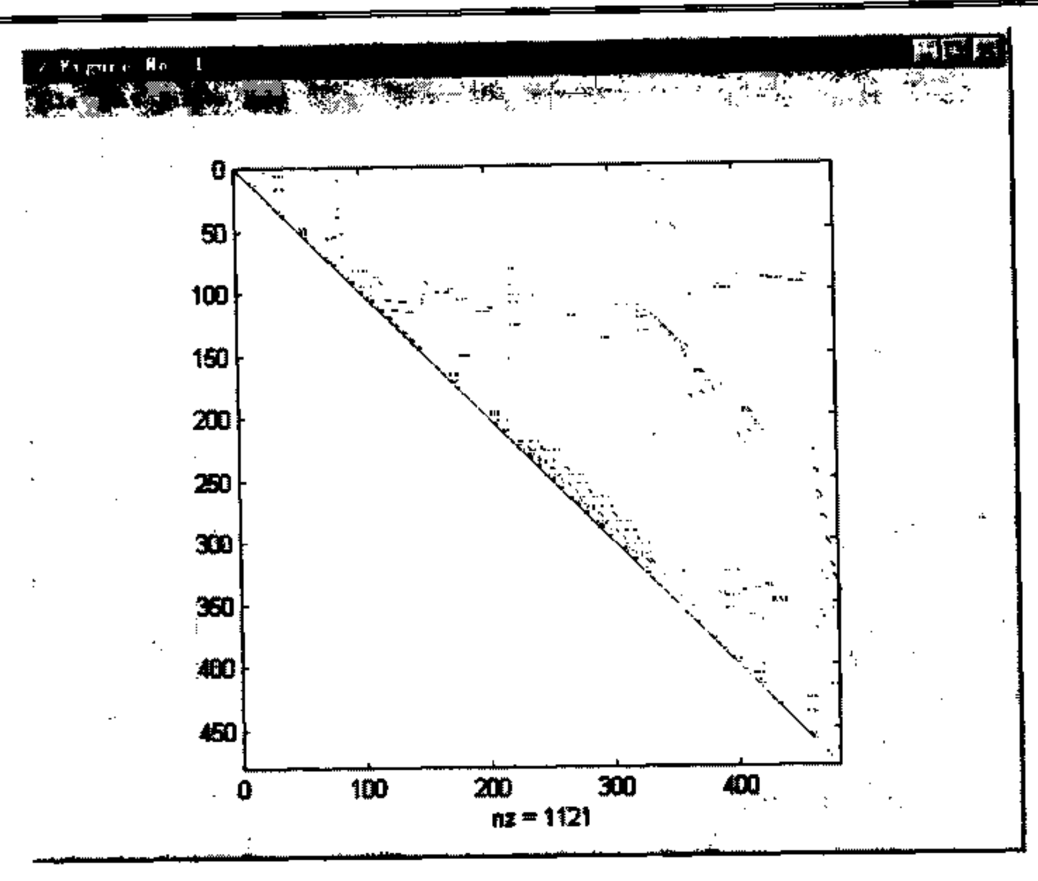


图 10-17 上三角矩阵 U 的结构图

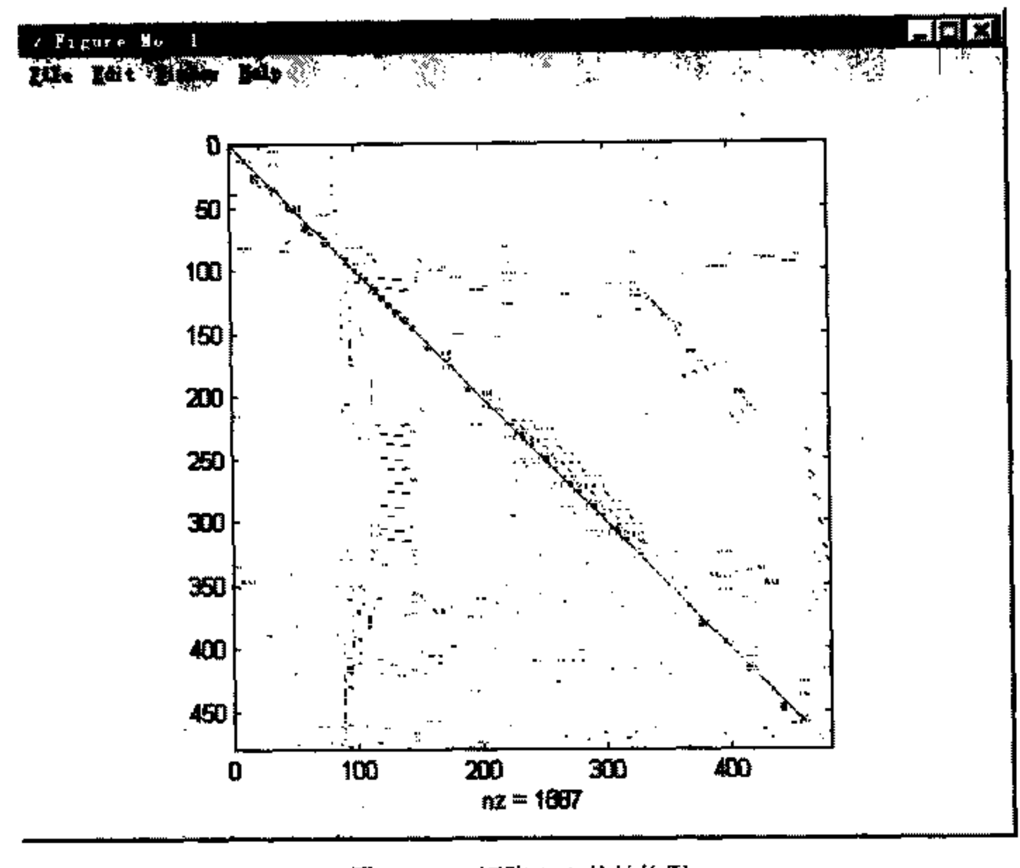


图 10-18 矩阵 P*S 的结构图

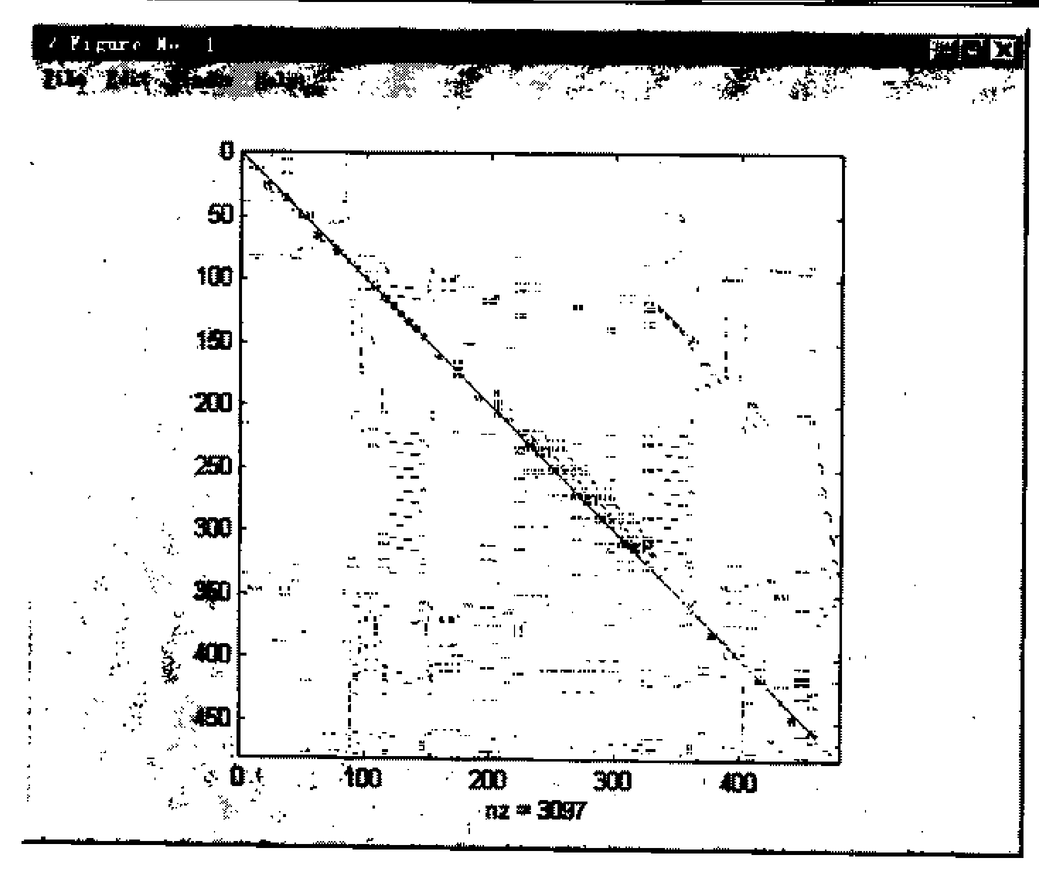


图 10-19　矩阵 L*U 的结构图

8. 预处理共轭梯度法

名称：pcg

预处理共轭梯度法。

语法：该函数有如下几种表达形式：

- x = pcg(A,b)
- pcg(A,b,tol)
- pcg(A,b,tol,maxit)
- pcg(A,b,tol,maxit,M)
- pcg(A,b,tol,maxit,M1,M2)
- pcg(A,b,tol,maxit,M1,M2,x0)
- x = pcg(A,b,tol,maxit,M1,M2,x0)
- [x,flag] = pcg(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres] = pcg(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter] = pcg(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter,resvec] = pcg(A,b,tol,maxit,M1,M2,x0)

描述：函数 pcg 将从一个初始估计值(缺省时为一个长度为 n 的零向量)开始进行迭代，直到满足收敛限或达到最大允许迭代次数时，停止迭代。从数学上讲，就是当迭代值 x 所引起的相对残差 norm(b-A*x)/norm(b)小于或等于给定的收敛限时，停止迭代。该收敛限的缺省值为 1e-6。缺省的最大允许迭代次数为 n 和 20 二者之间的最小者。

x = pcg(A,b)返回线性方程组系统 A*x = b 的解 x。其中 A 必须是对称正定矩阵，并为 n×n

阶方阵，同时列向量 b 的长度为 n。

当 A 不能够被显式地表达为矩阵时，用户可以将 A 表示为一个算子 afun，其中函数 afun(x) 将返回矩阵与向量的内积 A*x，而函数 afun(x,'transp')将返回 A'*x。该算子既可以是一个 M 文件名，也可以是 MATLAB 的一个内置函数。在这种情况下，n 将被取为列向量 b 的长度。

pcg(A,b,tol)返回线性方程组系统 A*x = b 的解 x。其中，参数 tol 用来指定迭代收敛限。

pcg(A,b,tol,maxit) 返回线性方程组系统 A*x = b 的解 x。其中，参数 tol 用来指定迭代收敛限，参数 maxit 用来指定最大允许迭代次数。

pcg(A,b,tol,maxit,M)和 pcg(A,b,tol,maxit,M1,M2)通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中，参数 tol 用来指定迭代收敛限，参数 maxit 用来指定最大允许迭代次数。

pcg(A,b,tol,maxit,M1,M2,x0)通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中，参数 tol 用来指定迭代收敛限，参数 maxit 用来指定最大允许迭代次数，参数 x0 用来指定初始估计值，若给定一个空矩阵[ ]，则缺省的初始估计值中全部为零向量。

x = pcg(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x。如果函数 pcg 迭代收敛，则在屏幕上会出现一个相应的信息。如果函数 pcg 迭代不收敛或已经达到了最大允许迭代次数或因意外情况程序被终止，此时会在屏幕上显示出一个警告信息，并给出迭代结束或终止时的相对残差 norm(b-A*x)/norm(b)和已经完成了的迭代次数。

[x,flag] = pcg(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 pcg 收敛性的参量 flag。该参量的取值和意义如表 10-4 中所示。

表 10-4　　参量 flag 的取值和意义

Flag	意义
0	Pcg 函数正常收敛。
1	Pcg 函数的迭代次数已经达到了最大允许迭代次数，但还没有收敛。
2	形如 M*y = r 的方程组中的某个方程是病态的，不能够采用‘\’运算符计算得到一个可用解。
3	迭代非正常终止(两个连续的迭代步得到了完全相同的结果)。
4	函数 pcg 迭代中得到的某一个标量太大或太小，从而使得计算终止。

[x,flag,relres] = pcg(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 pcg 收敛性的参量 flag。该参量的取值和意义如表 4 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)。如果 flag = 0，则 relres≤tol。

[x,flag,relres,iter] = pcg(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 pcg 收敛性的参量 flag。该参量的取值和意义如表 4 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter，而且满足 0≤iter≤maxit。

[x,flag,relres,iter,resvec] = pcg(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 pcg 收敛性的参量 flag。该参量的取值和意义如表 4 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter 以及在每一迭代步中得到的残差范数。

举例：

调用如下所示的函数得到矩阵 A 和向量 b：

```
A = delsq(numgrid('C',25))
b = ones(length(A),1)
```

现在用函数 pcg 求解方程组 A*x = b，在 MATLAB 命令窗口中输入如下命令：

```
[x,flag] = pcg(A,b)
```

结果为：

```
flag
    = 1
```

此时参数 flag 的取值为 1，表明函数 bicg 迭代了缺省的 20 步之后还没有收敛，因此自动终止计算。

如果事先调用函数 R = cholinc(A,1e-3)，对矩阵 A 进行不完全 Cholesky 分解，然后再调用函数[x2,flag2,relres2,iter2,resvec2] = pcg(A,b,1e-8,10,R',R)来求解方程组 A*x = b，可得：

```
flag2 = 0
relres2 =1.2059e-009
iter2 = 6
```

参数 flag2 的取值为 0，iter2 的取值为 6，这表明函数 pcg 在第 6 个迭代步收敛。

调用函数 semilogy(0:iter2,resvec2/norm(b),'-o')，将迭代过程中各步求得的解显示出来，如图 10-20 所示。

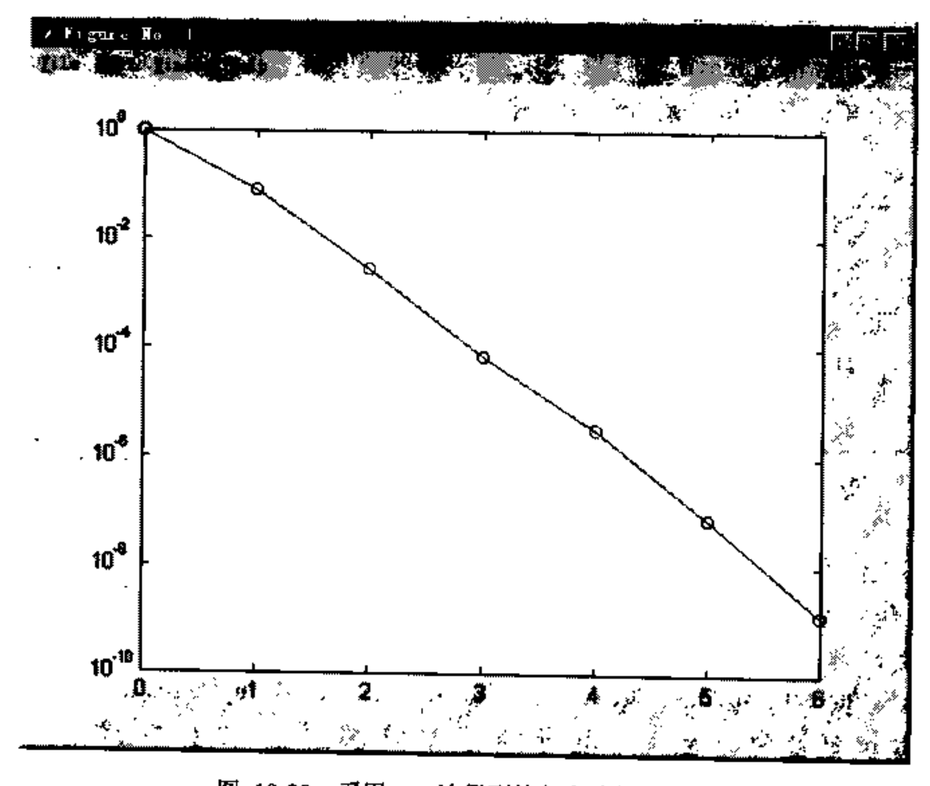


图 10-20　采用 pcg 法得到的各个迭代步的解

9. Quasi-Minimal 残差法

名称：qmr

Quasi-Minimal 残差法。

语法：该函数有如下几种表达式形式：

- x = qmr(A,b)
- qmr(A,b,tol)
- qmr(A,b,tol,maxit)
- qmr(A,b,tol,maxit,M1)
- qmr(A,b,tol,maxit,M1,M2)
- qmr(A,b,tol,maxit,M1,M2,x0)
- x = qmr(A,b,tol,maxit,M1,M2,x0)
- [x,flag] = qmr(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres] = qmr(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter] = qmr(A,b,tol,maxit,M1,M2,x0)
- [x,flag,relres,iter,resvec] = qmr(A,b,tol,maxit,M1,M2,x0)

描述：函数 qmr 将从一个初始估计值(缺省时为一个长度为 n 的零向量)开始进行迭代，直到满足收敛限或达到最大允许迭代次数时，停止迭代。从数学上讲，就是当迭代值 x 所引起的相对残差 norm(b-A*x)/norm(b)小于或等于给定的收敛限时，停止迭代。该收敛限的缺省值为 1e-6。缺省的最大允许迭代次数为 n 和 20 二者之间的最小者。

x = qmr(A,b)返回线性方程组系统 A*x = b 的解 x。其中 A 必须是 n×n 阶方阵，同时列向量 b 的长度为 n。

当A不能够被显式地表达为矩阵时，用户可以将A表示为一个算子afun，其中函数afun(x)将返回矩阵与向量的内积 A*x，而函数 afun(x,'transp')将返回 A'*x。该算子既可以是一个 M 文件名，也可以是 MATLAB 的一个内置函数。在这种情况下，n 将被取为列向量 b 的长度。

qmr(A,b,tol)返回线性方程组系统 A*x = b 的解 x。其中，参数 tol 用来指定迭代收敛限。

qmr(A,b,tol,maxit) 返回线性方程组系统 A*x = b 的解 x。其中，参数 tol 用来指定迭代收敛限，参数 maxit 用来指定最大允许迭代次数。

qmr(A,b,tol,maxit,M)和 qmr(A,b,tol,maxit,M1,M2)通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中，参数 tol 用来指定迭代收敛限，参数 maxit 用来指定最大允许迭代次数。

qmr(A,b,tol,maxit,M1,M2,x0)通过在形如 inv(M)*A*x = inv(M)*b 的方程两边左乘 M 或 M=M1*M2 来有效地得到该方程的解。其中，参数 tol 用来指定迭代收敛限，参数 maxit 用来指定最大允许迭代次数，参数 x0 用来指定初始估计值，若给定一个空矩阵[ ]，则缺省的初始估计值中全部为零向量。

x = qmr(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x。如果函数 qmr 迭代收敛，则在屏幕上会出现一个相应的信息。如果函数 qmr 迭代不收敛或已经达到了最大允许迭代次数或因意外情况程序被终止，此时会在屏幕上显示出一个警告信息，并给出迭代结束或终止时的相对残差 norm(b-A*x)/norm(b)和已经完成了的迭代次数。

[x,flag] = qmr(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 qmr 收敛性的参量 flag。该参量的取值和意义如表 10-5 中所示。

[x,flag,relres] = qmr(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 qmr 收敛性的参量 flag。该参量的取值和意义如表 10-5 中所示。同时还返回一个计算得到的相对

残差 norm(b-A*x)/norm(b)。如果 flag = 0，则 relres≤tol。

表 10-5　　参量 flag 的取值和意义

Flag	意义
0	Qmr 函数正常收敛。
1	Qmr 函数的迭代次数已经达到了最大允许迭代次数，但还没有收敛。
2	形如 M*y = r 的方程组中的某个方程是病态的，不能够采用‘\’运算符计算得到一个可用解。
3	迭代非正常终止(两个连续的迭代步得到了完全相同的结果)。
4	函数 qmr 迭代中得到的某一个标量太大或太小，从而使得计算终止。

[x,flag,relres,iter] = qmr(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 qmr 收敛性的参量 flag。该参量的取值和意义如表 10-5 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter，而且满足 0≤iter≤maxit。

[x,flag,relres,iter,resvec] = qmr(A,b,tol,maxit,M1,M2,x0)返回求解得到的解 x 和一个描述函数 qmr 收敛性的参量 flag。该参量的取值和意义如表 5 中所示。同时还返回一个计算得到的相对残差 norm(b-A*x)/norm(b)和已经完成的迭代数 iter 以及在每一迭代步中得到的残差范数。

举例：

调用如下所示的函数得到矩阵 A 和向量 b：

load west0479

A = west0479

b = sum(A,2)

现在用函数 qmr 求解方程组 A*x = b，在 MATLAB 命令窗口中输入如下命令：

[x,flag] = qmr(A,b)

结果为：

flag

　= 1

参数 flag 的取值为 1，表明函数 qmr 经过缺省的 20 步迭代后还没有收敛到缺省的收敛限 1e-6。

若首先调用函数[L1,U1] = luinc(A,1e-5)对 A 矩阵进行分解，然后再调用函数[x1,flag1] = qmr(A,b,1e-6,20,L1,U1)对方程组 A*x = b 进行求解，则可得到：

flag1 =

　　2

参数 flag1 的取值为 2。这是由于上三角矩阵 U1 的某个对角线元素的值为 0，因此当函数 bicgstab 在第一步中采用‘\’运算符求解方程 U1*y = r 时，该迭代步就被终止了。

现在提高 A 矩阵的分解精度，即调用函数[L2,U2] = luinc(A,1e-6)，然后再调用函数[x2,flag2,relres2,iter2,resvec2] = qmr(A,b,1e-15,10,L2,U2)对方程组 A*x = b 进行求解，则可得到：

flag2 =

　　0

relres2 =

2.0287e-016

iter2 =

8

参数 flag2 的取值为 0，iter2 的取值为 8，这表明函数 qmr 在第 8 个迭代步收敛。

调用函数 semilogy(0:iter2,resvec2/norm(b),'-o')，将迭代过程中各步求得的解显示出来，如图 10-21 所示。

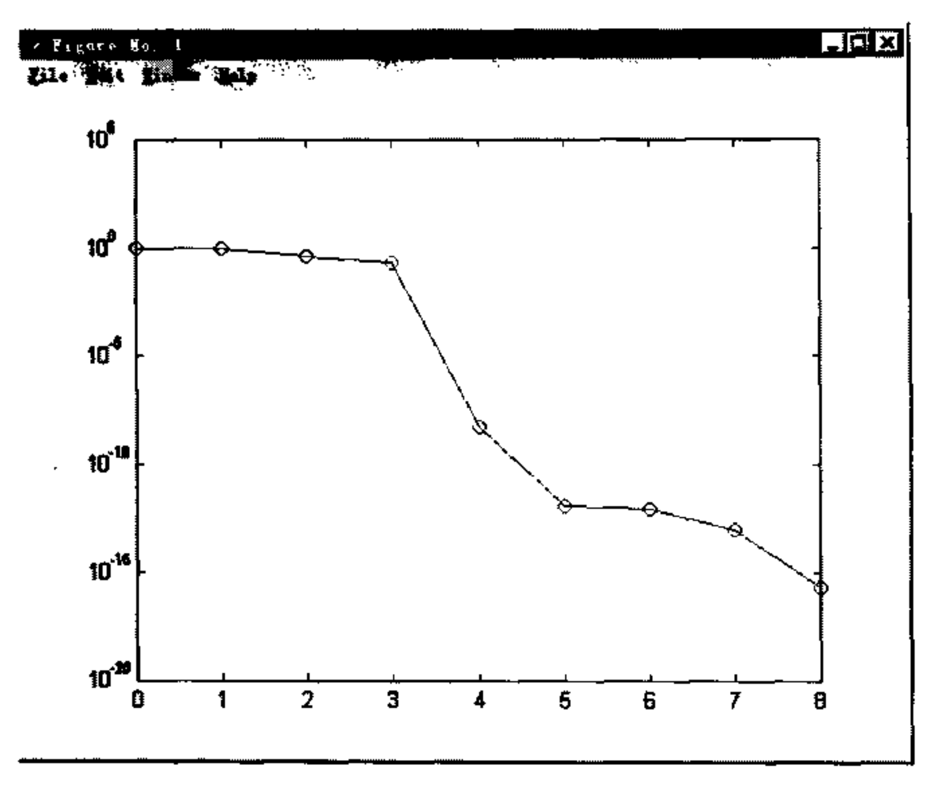


图 10-21　采用 Quasi-Minimal 残差法计算得到的解

10.8　稀疏矩阵的特征值和奇异值

1. 少数特征值和特征向量

名称：eigs

求少数特征值和特征向量。

语法：该函数有如下几种表达形式：

- d = eigs(A)
- d = eigs('Afun',n)
- d = eigs(A,B,k,sigma,options)
- d = eigs('Afun',n,B,k,sigma,options)
- [V,D] = eigs(A,...)
- [V,D] = eigs('Afun',n,...)
- [V,D,flag] = eigs(A,...)

- [V,D,flag] = eigs('Afun',n,...)

描述：函数 eigs 可以用来求解特征值问题 A*v = lambda*v 或广义特征值问题 A*v = lambda*B*v，其中 B 是一个对称正定矩阵。与函数 eigs 可以计算全部特征值和特征向量相比，函数 eigs 只能用来计算少数特征值和特征向量。其调用格式为：

d = eigs(A)求解特征值问题。其中输入参数为 n×n 阶方阵 A，A 可以为满阵，也可以为稀疏矩阵，可以是对称矩阵，也可以是非对称矩阵，可以是实数矩阵，也可以是复矩阵。输出项 d 为包含 k (n 和 6 之间的小者)个特征值的向量。

d = eigs('Afun',n)求解特征值问题。其中输入参数 Afun 是一个 M 文件名，该 M 文件对给定矩阵的列向量进行线性处理，第二个参数 n 为所求解问题的阶数。输出项 d 为包含 k (n 和 6 之间的小者)个特征值的向量。

d = eigs(A,B,k,sigma,options) 求解特征值问题。B 是一个和矩阵 A 具有同样尺寸的对称正定矩阵(n×n 阶)。如果 B 没有给定，则 MATLAB 自动令 B = eye(size(A))。参数 k(整数)指定希望得到的特征值的数目，如果 k 没有被给定，则 MATLAB 会将 n 和 6 之间的小者赋予 k。参量 sigma 是一个标量或包含两个字符的字符串，其可能的取值和对应的意义为：

如果 sigma 没有给定，则 MATLAB 求解 k 个模量最大的特征值。

如果 sigma = 0，则 MATLAB 求解 k 个模量最小的特征值。

如果 sigma 是一个实数或复数，则 MATLAB 求解离 sigma 最近的 k 个特征值。

如果 sigma 是包含两个字符的字符串，则：

取 ‘lm’ (缺省情况)，表示要求解 k 个模量最大的特征值。

取 ‘sm’，表示要求解 k 个模量最小的特征值。

取 ‘lr’，表示要求解 k 个实部最大的特征值。

取 ‘sr’，表示要求解 k 个实部最小的特征值。

取 ‘be’，表示要求解 k/2 个模量最大的特征值和 k/2 个模量最小的特征值。

结构 options 中所包含的参数及取值如表 10-6 所示。

表 10-6 结构 options 中所包含的参数及取值

参数	描述	缺省值
options.tol	指定收敛允许限，满足 norm(A*V-V*D) <= tol*norm(A)	A 为对称时，取 1e-10 A 非对称时，取 1e-6
Options.p	Arnoldi 要素的维数。	2*k
Options.maxit	最大允许迭代数。	300
Options.disp	在每一迭代步中显示的特征值数目。取 0 表示没有中间输出	20
Options.issym	当 Afun 为对称时，该参数取正数。	0
Options.cheb	如果 A 为字符串，sigma 取'lr'或'sr'，则该参数取正数。	0
Options.v0	Arnoldi 分解的初始向量。	rand(n,1)-.5

d = eigs('Afun',n,B,k,sigma,options)求解特征值问题。输入参数 Afun 是一个 M 文件名，该 M 文件对给定矩阵的列向量进行线性处理，第二个参数 n 为所求解问题的阶数。

[V,D] = eigs(A,...)和[V,D] = eigs('Afun',n,...)的输出参量 V 为一个包含 k 列的矩阵，D 是一个 k×k 阶对角矩阵，满足 A*V = V*D 或 A*V = B*V*D。

[V,D,flag] = eigs(A,...)和[V,D,flag] = eigs('Afun',n,...)的输出参数 flag 用来表示所求解的特征值问题是否收敛。flag = 0，表示收敛，flag = 1，表示不收敛。

举例：

MATLAB 中提供的 west0479 为一个 479×479 阶稀疏矩阵。该矩阵既有实数特征值，也有复共轭特征值对。本例将求解该矩阵的特征值问题。

为此，首先调用函数 load west0479，并调用函数 d = eigs(west0479)求解矩阵 west0479 的 6 个特征值，结果为：

```
d =
   1.0e+003 *
     0.0000 - 1.7007i
     0.0000 + 1.7007i
     0.1081 - 0.0541i
     0.1081 + 0.0541i
    -0.1009 - 0.0666i
    -0.1009 + 0.0666i
```

调用函数 dlm = eigs(west0479,8，'sm')，求解矩阵 west0479 的 8 个模量最小的特征值，结果为：

```
dlm =
     0.0002
    -0.0003
    -0.0004 + 0.0057i
    -0.0004 - 0.0057i
     0.0034 + 0.0168i
     0.0034 - 0.0168i
    -0.0211
     0.0225
```

2. *少数奇异值*

名称：svds

求少数奇异值。

语法：该函数有如下几种表达形式：

- s = svds(A)
- s = svds(A,k)
- s = svds(A,k,0)
- [U,S,V] = svds(A,...)

描述：s = svds(A)计算矩阵 A 的 5 个最大的奇异值和相应的奇异向量。输出量 s 为包含奇异值的向量。

s = svds(A,k)计算矩阵 A 的 k 个最大的奇异值和相应的奇异向量。输出量 s 为包含奇异值的向量。

s = svds(A,k,0)计算矩阵 A 的 k 个最小的奇异值和相应的奇异向量。输出量 s 为包含奇异值的向量。

[U,S,V] = svds(A,...)如果 A 为 m×n 阶矩阵，则 U 为由标准正交向量组成的 m×k 阶矩阵，S 为一个 k×k 阶对角阵，V 为由标准正交向量组成的 n×k 阶矩阵。

举例：

MATLAB 中提供的 west0479 为一个 479×479 阶稀疏矩阵。本例将求解该矩阵的奇异值问题。

为此，首先调用函数 load west0479 和函数 sl = svds(west0479)，求解矩阵 west0479 的 5 个最大的奇异值，结果为：

```
sl =
   1.0e+005 *
      3.1895
      3.1725
      3.1695
      3.1685
      3.1669
      0.3038
```

调用函数 sl = svds(west0479,5,0)，求解矩阵 west0479 的 5 个最小的奇异值，结果为：

```
sl =
   1.0e-004 *
      0.5169
      0.4505
      0.4020
      0.0424
      0.0098
```

总结起来，可以采用以下 3 种方法来计算矩阵 west0479 的最大奇异值：

svds(west0479,1) = 3.1895e+005；

max(svd(full(west0479))) = 3.1895e+005;

norm(full(west0479)) = 3.1895e+005。

或者利用函数 normest(west0479)来估计矩阵 west0479 的最大奇异值，结果为：

normest(west0479) = 3.1894e+005。

10.9 杂项函数

1．稀疏矩阵程序的参数

名称：spparms

设置稀疏矩阵程序的参数。

语法：该函数有如下几种表达形式：

- spparms('key',value)
- spparms
- values = spparms
- [keys,values] = spparms

- spparms(values)
- value = spparms('key')
- spparms('default')
- spparms('tight')

描述：spparms('key',value)设置稀疏线性方程运算符'\'、'/'以及函数 colmmd 和 symmmd 中包含的一个或多个可调参数。其中参数 key 的取值及其意义为：

取'spumoni'，为稀疏监控标记。当对应的值为 0 时(缺省情况)，没有诊断输出。当值为 1 时，会得到关于所选算法和存储情况的信息。当值为 2 时，会得到关于最小度算法的信息。

取'thr_rel'，'thr_abs'时，最小度极限值为 thr_rel*mindegree+thr_abs。

取'exact_d'，用来描述最小度的精度。取非零值，表示在最小度中采用精确度数；取 0 时，表示在最小度中采用近似度数。

取'supernd'，用来控制最小度计算过程。若取正数，则每隔 supernd 步，最小度与超节点进行合并。

取'rreduce'，用来控制最小度计算过程。若取正数，则每隔 rreduce 步，最小度要进行一次行分解。

取'wh_frac'，用来指定矩阵密度。在函数 colmmd 中，density > wh_frac 的行将被忽略。

取'autommd'，用来指定最小度的排序方式。取非 0 数时，表示要采用运算符 '\' 和 '/' 对最小度进行排序。

取'aug_rel'，'aug_abs'，为扩展方程 aug_rel*max(max(abs(A))) + aug_abs 的残差量。

spparms 返回关于当前设置的描述。

values = spparms 返回一个包含当前设置中各个参数取值的向量。

[keys,values] = spparms 返回的向量 values 中包含当前设置中各个参数的取值，同时还返回一个字符矩阵，该矩阵中的行向量就是各个参数的关键字。

spparms(values)利用给定的向量 values 中的数值对当前的参数进行设置。

value = spparms('key')返回某个参数的当前设置值。

spparms('default')对所有的参数采用缺省设置。

spparms('tight')对最小度排序参数采用严格设置方式。表 10-7 中列出了各个参数的缺省设置值和严格设置值。

表 10-7　　各参数的缺省设置值和严格设置值

关键字	缺省设置值	严格设置值
spumoni	0.0	
thr_rel	1.1	1.0
thr_abs	1.0	0.0
exact_d	0.0	1.0
Supernd	3.0	1.0
Rreduce	3.0	1.0
wh_frac	0.5	0.5
Autommd	1.0	
aug_rel	0.001	
aug_abs	0.0	

举例：

在 MATLAB 命令窗口中输入 spparms，可返回关于当前缺省设置的描述，如下所示：

```
No SParse MONItor output.
mmd: threshold = 1.1 * mindegree + 1,
     using approximate degrees in A'*A,
     supernode amalgamation every 3 stages,
     row reduction every 3 stages,
     withhold rows at least 50% dense in colmmd.
Minimum degree orderings used by \ and /。
```

调用函数 spparms('tight')，可对所有的参数采用严格设置。执行函数 spparms，可以将设置后的结果显示出来：

```
No SParse MONItor output.
mmd: threshold = 1 * mindegree + 0,
     using exact degrees in A'*A,
     supernode amalgamation every 1 stages,
     row reduction every 1 stages,
     withhold rows at least 50% dense in colmmd.
Minimum degree orderings used by \ and /。
```

第十一章　声音处理函数

11.1　常用声音函数

1. 音频信号转换

名称：lin2mu

把线性音频信号转换为 mu 法则。

语法：mu = lin2mu(y)

描述：mu = lin2mu(y)将音幅在[-1 1]范围内的音频信号转换为幅度范围为[0 255]、编码为 flints 的 mu 法则。

举例：

例如对于音幅为 0.5 的音频信号，调用函数 mu=lin2mu(0.5)，可以将该音频信号转换为 mu 法则，结果为：

```
mu =
    143
```

2. 音频信号线性化

名称：mu2lin

将 mu 法则音频信号线性化。

语法：y = mu2lin(mu)

描述：y = mu2lin(mu)将幅度范围为[0 255]的 8 位 mu 法则音频信号(编码为 flints)转换为幅度在[-s s]范围内的线性音频信号，其中 s = 32124/32768 ~= 0.9803。

举例：

例如对于幅度为 150 的 8 位 mu 法则音频信号，调用函数 y = mu2lin(150)可以将该音频信号线性化，结果为：

```
y =
     0.3944
```

3. 向量转换为声音

名称：sound

把向量转换为声音。

语法：该函数有如下几种表达形式：

- sound(y,Fs)
- sound(y)
- sound(y,Fs,bits)

描述：sound(y,Fs)把存储在向量 y 中的信号(抽样频率为 Fs)传送给计算机的扬声器。向

量 y 中元素的取值范围位于[-1.0 1.0]之内，超过该范围的值将被忽略。如果 y 是一个 n×2 阶矩阵，则利用函数 sound 就可以在支持立体音的计算机平台上播放出有立体效果的声音。

sound(y)以缺省的频率(或 8192 Hz)来播放声音。

sound(y,Fs,bits)采用 bits 位来播放声音。现在，大多数的计算机平台都支持 8 位或 16 位声音。

举例：

例如调用函数 sound([0.5 0.5])，就可以通过计算机的扬声器发出相应的声音。

4. 测量数据

名称： soundsc

测量数据并作为声音播放。

语法： 该函数有如下几种表达形式：

- soundsc(y,Fs)
- soundsc(y)
- soundsc(y,Fs,bits)
- soundsc(y,...,slim)

描述： soundsc(y,Fs)把存储在向量 y 中的信号(抽样频率为 Fs)传送给计算机的扬声器。其中 y 中元素的取值在播放之前应该转换为[-1.0 1.0]范围之内。

soundsc(y)以缺省的频率(或 8192 Hz)来播放声音。

soundsc(y,Fs,bits)采用 bits 位来播放声音。现在，大多数的计算机平台都支持 8 位或 16 位声音。

举例：

例如调用函数 soundsc([0.5 0.5])，就可以通过计算机的扬声器发出相应的声音。

11.2 特殊声音函数

1. 读入 NeXT/SUN 声音文件

名称： auread

读入 NeXT/SUN 声音文件(后缀为.au)。

语法： 该函数有如下几种表达形式：

- y = auread('aufile')
- [y,Fs,bits] = auread('aufile')
- [...] = auread('aufile',N)
- [...] = auread('aufile',[N1,N2])
- siz = auread('aufile','size')

描述： y = auread('aufile')读入由 aufile 指定的一个声音文件，并将数据返回到 y 中。

[y,Fs,bits] = auread('aufile')还返回频率值和声音位数。

[...] = auread('aufile',N)仅仅返回声音文件中每个频道的前 N 个数据。

[...] = auread('aufile',[N1,N2])仅仅返回声音文件中每个频道的第 N1 到 N2 个数据。

siz = auread('aufile','size')返回声音数据文件的大小。

2. 写 NeXT/SUN 声音文件

名称：auwrite

写 NeXT/SUN 声音文件(后缀为.au)。

语法：该函数有如下几种表达形式：

- auwrite(y,'aufile')
- auwrite(y,Fs,'aufile')
- auwrite(y,Fs,N,'aufile')
- auwrite(y,Fs,N,'method','aufile')

描述：auwrite(y,'aufile')将描述声音的数据写入到 aufile 指定的数据文件。

auwrite(y,Fs,'aufile')还指定声音的频率。

auwrite(y,Fs,N,'aufile')选择编码器的位数。

auwrite(y,Fs,N,'method','aufile')允许选择编码的方法。

11.3 WAV 声音函数

1. 读入 WAV 声音文件

名称：wavread

写 WAV 声音文件(后缀为.wav)。

语法：该函数有如下几种表达形式：

- y = wavread('filename')
- [y,Fs,bits] = wavread('filename')
- [...] = wavread('filename',N)
- [...] = wavread('filename',[N1 N2])
- [...] = wavread('filename','size')

描述：该函数支持 8 位或 16 位多频道 WAVE 数据。

y = wavread('filename')读入由 filename 指定的一个 WAV 声音文件，并将数据返回到 y 中。

[y,Fs,bits] = wavread('filename')还返回频率值和声音位数。

[...] = wavread('filename',N)仅仅返回声音文件中每个频道的前 N 个数据。

[...] = wavread('filename',[N1 N2])仅仅返回声音文件中每个频道的第 N1 到 N2 个数据。

siz = wavread('aufile','size')返回声音数据文件的大小。

2. 写 WAV 声音文件

名称：wavwrite

写 WAV 声音文件(后缀为.wav)。

语法：该函数有如下几种表达形式：

- wavwrite(y,'filename')
- wavwrite(y,Fs,'filename')

- wavwrite(y,Fs,N,'filename')

描述：该函数支持 8 位或 16 位多频道 WAVE 数据。

wavwrite(y,'filename')将描述声音的数据写入到 filename 指定的数据文件。

wavwrite(y,Fs,'filename')还指定声音的频率值。

wavwrite(y,Fs,N,'filename')选择编码器的位数。

第十二章　字符串函数

12.1　常用函数

1. 绝对值或复数的模

名称：abs

求解绝对值或复数的模。

语法：Y = abs(X)

描述：Y = abs(X)当 X 中的元素为实数时，返回各元素的绝对值所组成的向量 Y。当 X 中的元素为复数时，该函数返回复数的模，即 abs(X) = sqrt(real(X).^2 + imag(X).^2)。

举例：

例如：

abs(-5) = 5

abs(3+4i) = 5

2. 运行字符串所表示的表达式

名称：eval

运行字符串所表示的表达式。

语法：该函数有如下几种表达形式：

- eval(expression)
- [a1,a2,a3,...] = eval(expression)
- eval(expression,catch_expr)

描述：eval(expression)执行字符串 expression 所表示的 MATLAB 表达式。用户可以通过在方括弧中连接子字符串和变量来构造 expression，如下所示：

expression = [string1,int2str(var),string2,...]。

[a1,a2,a3,...] = eval(expression)执行字符串 expression 所表示的 MATLAB 表达式，并将结果返回到相应的变量中。

eval(expression,catch_expr)执行字符串 expression 所表示的 MATLAB 表达式，如果发现有错误，就执行 catch_expr 所表示的表达式。如果 expression 表达式运行有错误，利用函数 lasterr 就可以得到该错误信息。

举例：

利用函数 eval 执行一个简单的 MATLAB 表达式 A = '1+4'。

在 MATLAB 命令窗口中输入 A = '1+4'和 aval = eval(A)，结果为：

```
aval =

     5
```

执行以下函数可以产生 12 个矩阵，分别命名为 M1 到 M12：

```
for n = 1:12
        magic_str = ['M',int2str(n),' = magic(n)'];
        eval(magic_str)
end
```

得到的第 12 个矩阵 M12 为：

```
M12 =
   144     2     3   141   140     6     7   137   136    10    11   133
    13   131   130    16    17   127   126    20    21   123   122    24
    25   119   118    28    29   115   114    32    33   111   110    36
   108    38    39   105   104    42    43   101   100    46    47    97
    96    50    51    93    92    54    55    89    88    58    59    85
    61    83    82    64    65    79    78    68    69    75    74    72
    73    71    70    76    77    67    66    80    81    63    62    84
    60    86    87    57    56    90    91    53    52    94    95    49
    48    98    99    45    44   102   103    41    40   106   107    37
   109    35    34   112   113    31    30   116   117    27    26   120
   121    23    22   124   125    19    18   128   129    15    14   132
    12   134   135     9     8   138   139     5     4   142   143     1
```

3. 复数的实部

名称：real

复数的实部。

语法：X = real(Z)

描述：X = real(Z)返回复数数组 Z 中各个元素的实部。

举例：

例如执行函数 x = real(2+3*i)时，结果为：

```
x =
     2
```

4. MATLAB 字符串操作

名称：strings

MATLAB 字符串操作。

语法：该函数有如下几种表达形式：

S = 'Any Characters'

S = string(X)

X = numeric(S)

描述：S = 'Any Characters'返回向量 S，该向量的元素为给定字符对应的 ASCII 码，长度为字符的数目。若给定的字符串中包含引号，则该引号要用两个引号括起来。

S = string(X)将包含正整数的数组 X 转换为 MATLAB 的字符数组。

X = numeric(S)将字符串 S 转换为相应的数值代码。

举例：

例如执行函数 S = 'I love China'，结果为：

S =

I love China。

若执行函数 s = ['It is 1 o''clock', 7]，结果为：

s =

It is 1 o'clock。

若执行函数 S = string([89 86 76 79])，结果为：

S =

YVLO。

若执行函数 X = numeric(S)，结果为：

X =

73　32　108　111　118　101　32　67　104　105　110　97。

12.2　字符串操作

1．去掉字符串末尾的空格

名称：deblank

去掉字符串末尾的空格。

语法：该函数有如下两种表达形式：

- str = deblank(str)
- c = deblank(c)

描述：函数 deblank 对于清理由字符组成的行向量非常有用。

str = deblank(str)去掉字符串 str 末尾的空格。

c = deblank(c)如果 c 是一个字符串细胞数组，则对 c 的每一个元素作用函数 deblank。

举例：

例如运行函数 str='Tsinghua University　　'，结果为：

str=

　'Tsinghua University　　'

若调用函数 str = deblank(str)，结果为：

str=

　'Tsinghua University'

2．查找字符串

名称：findstr

查找字符串。

语法：k = findstr(str1,str2)

描述：k = findstr(str1,str2)在长字符串 str1 中查找短字符串 str2，并将找到的索引返回给向量 k。

举例：

例如对于字符串 str1 = 'Find the starting indices of the shorter string.'和字符串 str2 = 'the'，调用函数 k = findstr(str1,str2)，结果为：

k =

    6   30

3. 转换为小写

名称：lower

转换为小写。

语法：该函数有如下两种表达形式：

- t = lower('str')
- B = lower(A)

描述：t = lower('str')把字符串 str 中的所有大写字母转换为小写，原来的小写字母保持不变。

B = lower(A)如果 A 是一个字符串细胞数组，该函数将 A 中所有字符串的大写字母转换为小写字母，而原来的小写字母则保持不变，并将变换后的数组返回到一个和 A 具有相同尺寸的细胞数组 B。

举例：

例如对于字符串 str = 'Tsinghua University, P. R. China'，调用函数 t = lower('str')，对该字符串进行大小写转换，结果为：

t =

tsinghua university, p. r. china

4. 字符串组合

名称：strcat

字符串组合。

语法：t = strcat(s1,s2,s3,...)

描述：t = strcat(s1,s2,s3,...)对于字符串数组 s1、s2、s3…等进行水平组合。其中输入字符串数组 s1、s2、s3…必须有相同数目的行。

举例：

例如对于字符串 s1 = 'Tsinghua University,'，s2 = 'Department of Computer Sicence, '，s3 = 'P.R. China'，调用函数 t = strcat(s1,s2,s3)，将字符串进行组合，结果为：

t =

Tsinghua University,Department of Computer Sicence,P.R. China

5. 字符串比较

名称：strcmp

字符串比较。

语法：该函数有如下两种表达形式：

- k = strcmp('str1','str2')
- TF = strcmp(S,T)

描述：在 MATLAB 里，可以对字符矩阵和细胞数组里的字符串进行比较。可对两个字

符串或字符串中的子串进行比较，也可以对字符串中的单个字符做比较，还可把字符串中的每个元素分为字母和空格。其调用格式为：

k = strcmp('str1','str2')比较两个字符串 str1 和 str2 是否相等。如果相等，则返回 1；否则返回 0。

TF = strcmp(S,T)如果 S 或 T 为字符串细胞数组，则该函数返回一个和 S 与 T 尺寸相同的数组 TF，如果 S 和 T 中某一位置处的对应元素相同，则 TF 中相应位置处的元素取 1，否则取 0。

举例：

例如对于字符串 s1 = 'Tsinghua University'，s2 = 'Department of Computer Sicence, '，调用函数 k = strcmp(s1,s2)对这两个字符串进行比较，结果为：

```
k =
     0
```

若将上面两个字符串改为 s1 = 'Tsinghua University'和 s2 = 'Tsinghua University'，再调用函数 k = strcmp(s1,s2)对这两个字符串进行比较，结果为：

```
k =
     1
```

利用函数 A = {'MATLAB';'Simulink';'The MathWorks'}生成一个字符串细胞数组：

```
A =
    'MATLAB'
    'Simulink'
    'The MathWorks'
```

调用函数 strcmp('The MathWorks',A)进行比较，结果为：

```
ans =
     0
     0
     1
```

再如，若运行函数 strcmp('hello', {'hello','world'})，结果为：

```
ans =
     1     0
```

若运行函数 strcmp({'hello'}, ['hello';'world'])，结果为：

```
ans =
     1
     0
```

下面将字符串与细胞数组进行比较：

在 MATLAB 命令窗口中输入 strcmp('hello ', {'hello','world'})，可得：

```
ans =
     0     0
```

若输入 strcmp(deblank('hello '), {'hello','world'})，结果为：

```
ans =
```

1　　0

6. 忽略大小写对字符串进行比较

名称：strcmpi

忽略大小写对字符串进行比较。

语法：该函数有如下两种表达形式：

- strcmpi(str1,str2)
- strcmpi(S,T)

描述：strcmpi(str1,str2)如果字符串 str1 和 str2 中的字母除了大小写之外完全相同，则返回 1，否则返回 0。

strcmpi(S,T)如果 S 或 T 为字符串细胞数组，则该函数返回一个和 S 与 T 尺寸相同的数组，如果 S 和 T 中某一位置处的对应元素除了大小写之外完全相同，则返回数组中相应位置处的元素取 1，否则取 0。

另外，函数 strcmpi 还支持非英文字母组成的字符串之间的比较。

举例：

例如对于字符串 s1 = 'Tsinghua University'，s2 = 'TSINGHUA UNIVERSITY'，调用函数 strcmpi(s1,s2)对这两个字符串进行比较，结果为：

```
ans =
     1
```

此时若调用函数 strcmp(s1,s2)，则比较的结果为：

```
ans =
     0
```

利用函数 A = {'MATLAB';'Simulink';'The MathWorks'}生成一个字符串细胞数组：

```
A =
    'MATLAB'
    'Simulink'
    'The MathWorks'
```

调用函数 strcmpi('THE MATHWORKS',A)进行比较，结果为：

```
ans =
     0
     0
     1
```

此时若调用函数 strcmp('THE MATHWORKS',A)，则比较的结果为：

```
ans =
     0
     0
     0
```

再如，执行函数 strcmpi('中华人民共和国','中国')，可得：

```
ans =
     0
```

执行 strcmpi('中华人民共和国','中华人民共和国')，可得：

ans =

1

7. 调整字符串排列位置

名称：strjust

调整字符串排列位置。

语法：该函数有如下几种表达形式：

- T = strjust(S)
- T = strjust(S,'right')
- T = strjust(S,'left')
- T = strjust(S,'center')

描述：T = strjust(S)将字符串数组 S 中的元素右对齐。

T = strjust(S,'right')将字符串数组 S 中的元素右对齐。

T = strjust(S,'left')将字符串数组 S 中的元素左对齐。

T = strjust(S,'center')将字符串数组 S 中的元素居中对齐。

举例：

例如对于字符串 s = '　　Tsinghua University　　'，调用函数 T = strjust(s)或 T=strjust(s,'right')，将该字符串右对齐，结果为：

T =

'　　　　Tsinghua University'

若调用函数 T = strjust(s,'left')，则可将该字符串左对齐，结果为：

T =

'Tsinghua University　　　　'

若调用函数 T = strjust(s,'center')，则可将该字符串居中对齐，结果为：

T =

'　　Tsinghua University　　'

8. 寻找符合条件的行

名称：strmatch

寻找符合条件的行。

语法：该函数有如下两种表达形式：

- i = strmatch('str',STRS)
- i = strmatch('str',STRS,'exact')

描述：i = strmatch('str',STRS)在字符串数组或字符串细胞数组 STRS 的行向量中寻找以字符串 str 开头的字符串，并返回相应行的索引。

i = strmatch('str',STRS,'exact')在字符串数组或字符串细胞数组 STRS 的行向量中寻找与字符串 str 精确匹配的字符串，并返回相匹配行的索引。

举例：

例如在 MATLAB 命令窗口中执行 i = strmatch('max',strvcat('max','minimax','maximum'))，可得：

i =

1

3

这是由于给定字符串数组中的第 1 和第 3 行均以字符串 max 开头。

若执行 i = strmatch('max',strvcat('max','minimax','maximum'),'exact')，结果为：

i =

1

这是由于在给定字符串数组中，只有第 1 行的元素与字符串 max 精确匹配。

9. 比较字符串的前 n 个字符

名称：strncmp

比较字符串的前 n 个字符。

语法：该函数有如下两种表达形式：

- k = strncmp('str1','str2',n)
- TF = strncmp(S,T,n)

描述：k = strncmp('str1','str2',n)如果字符串 str1 和 str2 中的前 n 个字符相同，则返回 1，否则，返回 0。其中 str1 和 str2 也可以是字符串细胞数组。

TF = strncmp(S,T,n)如果 S 或 T 为字符串细胞数组，则该函数返回一个和 S 与 T 尺寸相同的数组 TF，如果 S 和 T 中某一位置处的对应元素的前 n 个字符相同，则 TF 中相应位置处的元素取 1，否则取 0。

举例：

利用函数 A={'me';'you';'he'}和 B={'me';'yourself';'it'}生成两个字符串细胞数组：

A =

'me'

'you'

'he'

B =

'me'

'yourself'

'it'

然后调用函数 strncmp(A,B,1)，结果为：

ans =

1

1

0

若调用函数 strncmp(A,B,3)，结果为：

ans =

0

1

0

10. 查找和替换

名称：strrep

字符串的查找和替换。

语法：str = strrep(str1,str2,str3)

描述：在 MATLAB 里，函数 findstr 和 strrep 分别用来查找和替换字符串中的子串。其中函数 findstr 将给出子串在字符串中的位置，位置用子串的第一个字符所在的位置表示。函数 strrep 把字符串中的子串替换为新的子串，其调用格式为：

str = strrep(str1,str2,str3)将字符串 str1 中所有出现字符串 str2 的地方用字符串 str3 来替换。

若 str1、str2 和 str3 中的任意一个是字符串细胞数组，则函数 strrep(str1,str2,str3)返回一个和 str1、str2、str3 尺寸相同的字符串细胞数组。

举例：

例如对于字符串 s1 = 'This is a good example.'，调用函数 str = strrep(s1,'good','great')，可得：

```
str =
This is a great example.
```

利用函数 A={'MATLAB' 'SIMULINK';'Toolboxes' 'The MathWorks'}，B={'Handle Graphics' 'Real Time Workshop'; 'Toolboxes' 'The MathWorks'}和 C={'Singal Processing' 'Image Processing'; 'MATLAB' 'SIMULINK'}生成如下所示的三个字符串细胞数组：

```
A =
    'MATLAB'        'SIMULINK'
    'Toolboxes'     'The MathWorks'
B =
    'Handle Graphics'    'Real Time Workshop'
    'Toolboxes'          'The MathWorks'
C =
    'Singal Processing'    'Image Processing'
    'MATLAB'               'SIMULINK'
```

调用函数 strrep(A,B,C)，可得：

```
ans =
    'MATLAB'    'SIMULINK'
    'MATLAB'    'SIMULINK'
```

11. 查找位置

名称：strtok

查找某个字符最先出现的位置。

语法：该函数有如下几种表达形式：

- token = strtok('str',delimiter)
- token = strtok('str')
- [token,rem] = strtok(...)

描述：函数 strtok 用于返回字符串中某个特定字符前面的所有字符。其调用格式为：

token = strtok('str',delimiter)返回字符串 str 中第一次出现向量 delimiter 中所包含的字符的位置前面的所有字符。

token = strtok('str')返回字符串 str 中第一个空格键前出现的所有字符。

[token,rem] = strtok(...)在 token 中返回字符串 str 中第一次出现向量 delimiter 中所包含的字符的位置前面的所有字符，在 rem 中返回剩余的字符(包括空格键)。

举例：

例如对于字符串 s = 'This is a good example.'，调用函数 token = strtok(s,'x')，可以得到字符串 s 中第一次出现字符 x 前的所有字符，结果如下所示：

token =

This is a good e

若调用函数 token = strtok(s)，则可得：

token =

This

若调用函数[token,rem] = strtok(s)，可得：

token =

This

rem =

 is a good example.

12. 竖向组合

名称：strvcat

字符串的竖向组合。

语法：S = strvcat(t1,t2,t3,...)

描述：S = strvcat(t1,t2,t3,...)对于字符串数组 s1、s2、s3...等进行竖向组合。

举例：

例如执行函数 strvcat('Hello','Yes')得到的结果与执行['Hello';'Yes ']得到的结果相同，均为：

ans =

Hello

Yes

再如，对于字符串 t1 = 'first'，t2 = 'string'、t3 = 'matrix'和 t4 = 'second'，调用函数 S1 = strvcat(t1,t2,t3)可得：

S1 =

first

string

matrix

若调用函数 S2 = strvcat(t4,t2,t3)，则可得：

S2 =

second

string

matrix

若调用函数 S3 = strvcat(S1,S2)，则可得：

S3 =

first

string

matrix

second

string

matrix

13. 符号变量

名称：symvar

决定表达式中的符号变量。

语法：symvar('str')

描述：symvar('str')在字符串 str 中寻找除了 i、j、pi、inf、nan 和 eps 以及通用函数以外的其他标识符，并且将寻找结果以字符串细胞数组的形式返回。如果没有找到相应的变量，则返回一个空细胞数组{}。

举例：

例如执行函数 symvar('cos(pi*x - beta1)')，可得到：

{'beta1','x'}。

若执行函数 symvar('pi eps nan')，则返回：

{}。

14. TeX 格式

名称：texlabel

将字符串转换为 TeX 格式。

语法：该函数有如下两种表达形式：

- texlabel(f)
- texlabel(f,'literal')

描述：texlabel(f)将 MATLAB 表达式 f 转换为文本字符串中的等效格式。如果 f 表示的是希腊字符变量名(如 lambda、delta 等)，则返回真正的希腊字母。

texlabel(f,'literal')将希腊变量名转换为文字表达。

15. 大写

名称：upper

将字符串转换为大写。

语法：该函数有如下两种表达形式：

- t　= upper('str')
- B = upper(A)

描述：t　= upper('str')把字符串 str 中的所有小写字母转换为大写，原来的大写字母保持不变。

B = upper(A)如果 A 是一个字符串细胞数组，该函数将 A 中所有字符串的小写字母转换为大写字母，而原来的大写字母则保持不变，并将变换后的数组返回到一个和 A 具有相同尺寸的细胞数组 B。

举例：

例如对于字符串 str = 'Tsinghua University, P. R. China'，调用函数 t = upper(str)，对该字符串进行大小写转换，可以得到：

t =

TSINGHUA UNIVERSITY, P. R. CHINA

12.3 字符串和数值的转换

数值和字符串是两个区别很大的概念，但有时也颇为费解。数值 1 和字符串“1”在 MATLAB 里完全不同。如果某个变量的值为数值 1，MATLAB 保存该变量采用双精度型，占用 8 个字节；而当变量的值为字符串“1”时，在 MATLAB 中只占用 2 个字节，这两个字节用来保存 1 的 ASCII 的值。在 MATLAB 中，可以把数值转换为字符串。数值转换成的字符串可以是十进制的字符串，也可以是二进制和十六进制的字符串。

1. 字符数组

名称： char

生成字符数组。

语法： 该函数有如下几种表达形式：

- S = char(X)
- S = char(C)
- S = char(t1,t2.t3...)

描述： S = char(X)将包含正整数的数组 X 转换为 MATLAB 中的字符数组。

S = char(C)如果 C 为一个字符串细胞数组，则将 C 中每一个单元转换成字符数组 S 中的行。

S = char(t1,t2.t3...)返回一个以字符串 t1、t2、t3…为行向量的字符数组 S。

举例：

例如调用函数 S = char([34 99 44 88 47])，可以得到：

S =

"c,X/

再如，调用函数 ascii = char(reshape(32:127,32,3)')，可以得到：

ascii =

 !"#$%&'()*+,-./0123456789:;<=>?

@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_

`abcdefghijklmnopqrstuvwxyz{|}~•

2. 整数转换为字符串

名称： int2str

把整数转换为字符串。

语法：str = int2str(N)

描述：函数 int2str 把整数转换成字符串，字符串中的数字用十进制表示。函数名中的 2 和英文单词'to'同音，使用这类函数，可以望文生义。其调用格式为：

str = int2str(N)其中 N 可以是一个单独的整数，或者是一个由整数组成的向量或矩阵。

举例：

首先令 x = 5317，调用函数 whos 查看 x 的属性，可得：

```
  Name          Size              Bytes  Class
  x             1x1                   8  double array
  Grand total is 1 elements using 8 bytes
```

然后调用函数 y = int2str(x)，将该数值转换成字符矩阵：

```
y =
5317
```

调用函数 clear x 和 whos，查看 y 的属性，可得：

```
  Name          Size              Bytes  Class
  y             1x4                   8  char array
  Grand total is 4 elements using 8 bytes
```

再如，对于矩阵 X=eye(3)，即：

```
X =
      1     0     0
      0     1     0
      0     0     1
```

调用函数 whos 查看 X 的属性，可得：

```
  Name        Size            Bytes  Class
  X           3x3                72  double array
  Grand total is 9 elements using 72 bytes
```

调用函数 Y = int2str(X)，得到：

```
Y =
1  0  0
0  1  0
0  0  1
```

调用函数 clear X 和 whos 查看 X 的属性，可得：

```
  Name        Size            Bytes  Class
  Y           3x7                42  char array
  Grand total is 21 elements using 42 bytes
```

3. 矩阵转换为字符串

名称：mat2str

把矩阵转换为字符串。

语法：该函数有如下两种表达形式：

- str = mat2str(A)
- str = mat2str(A,n)

描述：str = mat2str(A)将矩阵 A 转换为字符串。

str = mat2str(A,n)将矩阵 A 转换为字符串(精确到 n 位)。

举例：

例如对于矩阵

```
A =
     1     2
     3     4
```

调用函数 whos 查看其属性，结果为：

```
  Name        Size            Bytes  Class
  A           2x2                32  double array
  Grand total is 4 elements using 32 bytes
```

然后调用函数 str = mat2str(A)进行转换，可得：

```
str =
[1 2;3 4]
```

再调用函数 clear A 和 whos 查看 str 的属性，结果为：

```
  Name        Size            Bytes  Class
  str         1x9                18  char array
  Grand total is 9 elements using 18 bytes
```

再如对于如下所示的矩阵

```
A =
    1.8889    2.5675
    3.5679    4.3548
```

调用函数 str = mat2str(A,2)进行转换，并精确到 2 位，可得：

```
str =
[1.9 2.6;3.6 4.4]
```

4. 数值转换为字符串

名称：num2str

把数值转换为字符串。

语法：该函数有如下几种表达形式：

- str = num2str(A)
- str = num2str(A,precision)
- str = num2str(A,format)

描述：函数 num2str 可以将数值转换成字符串，并可以指定输出的格式。其调用格式为：

str = num2str(A)将数组 A 转换为一个字符串表达式 str(精确到 4 位，如果需要也可以采用指数表达式)。

str = num2str(A,precision)用 precision 指定的精度将数组 A 转换为一个字符串表达式 str。参数 precision 的缺省值为 4。

str = num2str(A,format)用 format 指定的格式将数组 A 转换为一个字符串表达式 str。参量 format 的缺省值为'%11.4g'。

函数 int2str 和 num2str 对于图形的标注是很有用的。在图形标注中，需要使用字符串，而在该字符串中往往包含数字，这时可用 int2str 和 num2str 把其中的数字转换为字符串。

举例：

例如执行函数 num2str(pi)，可以得到：

```
ans =
3.1416
```

若执行函数 num2str(pi,10)，则可得：

```
ans =
3.141592654
```

若执行函数 num2str(pi,'%16.8g')，则可得：

```
ans =
3.1415927
```

下面的函数将画出一条平方曲线，并在 x 轴旁标注横坐标的取值范围：

```
x=1:0.2:10
plot(x,x.^2)
str1=num2str(min(x))
str2=num2str(max(x))
out = ['Value of f from' str1 'to' str2]
xlabel(out)
```

结果如图 12-1 所示。

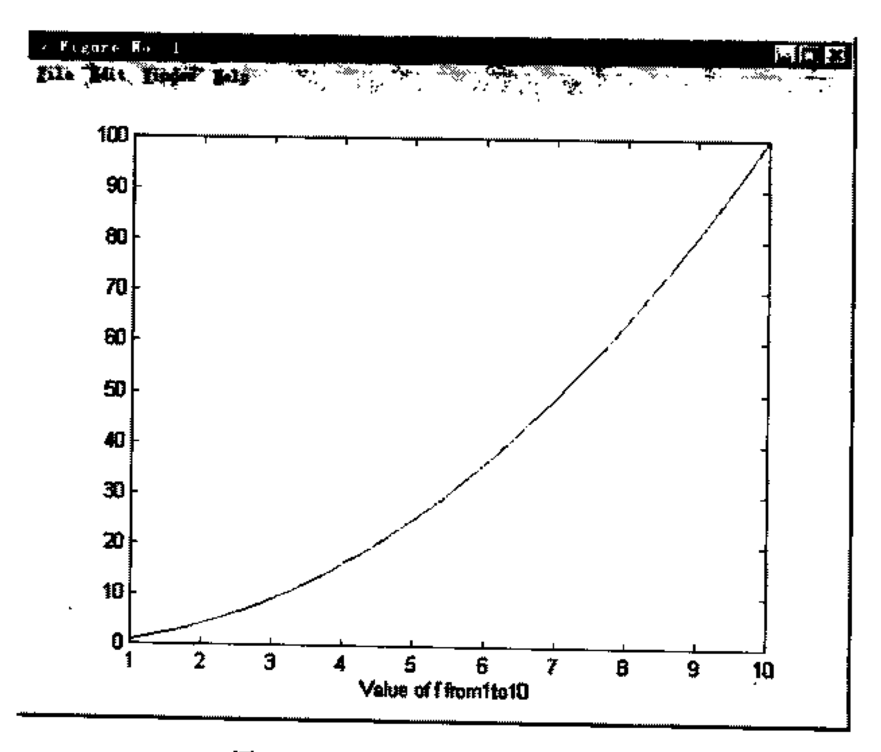


图 12-1　平方曲线图(并进行了标注)

5. 格式输出字符串

名称：sprintf

格式输出字符串。

语法：该函数有如下两种表达形式：

- s = sprintf(format,A,...)
- [s,errrmsg] = sprintf(format,A,...)

描述：s = sprintf(format,A,...)按照字符串 format 指定的格式对矩阵 A 中的数据进行处理，并将结果返回到 MATLAB 字符串变量 s 中。其中 format 用来指定输出格式的符号、对齐方式、有效位数等。

[s,errrmsg] = sprintf(format,A,...)按照字符串 format 指定的格式对矩阵 A 中的数据进行处理，并将结果返回到 MATLAB 字符串变量 s 中。如果出现错误，则在字符串 errmsg 中返回一个错误信息，如果没有出现错误，则返回空矩阵。

MATLAB 中的函数 sprintf 类似于 ANSI C 语言中的 sprintf()函数。表 1 中列出了 sprintf 函数的输出格式说明。

表 12-1　　sprintf 函数的输出格式说明

表示符	格式
%c	采用单字符格式。
%d	采用有符号十进制格式。
%e	采用指数表示法，并用小写的 e。
%E	采用指数表示法，并用大写的 E。
%f	采用固定位数输出格式。
%g	采用比%e 和%f 更紧凑的格式。
%G	与%g 相同，只是采用大写的 E。
%o	采用八进制格式。
%s	采用字符串格式。
%u	采用无符号的十进制格式。
%x	采用十六进制格式，而且使用小写字母 a-f。
%X	采用十六进制格式，而且使用大写字母 A-F。

表 12-2 中列出函数 sprintf 使用的一些特殊格式。

表 12-2　　sprintf 函数中使用的一些特殊格式

表示符	格式
\b	退后一格。
\f	换页。
\n	新行。
\r	回车。
\t	Tab 键。
\\	反斜线
\''或''	单引号。
%%	百分比符号。

举例：

执行函数 sprintf('%0.5g',(1+sqrt(5))/2)，可以得到：

ans =

1.618

执行函数 sprintf('%0.5g',1/eps)，可以得到：

ans =

4.5036e+015

执行函数 sprintf('%15.5f',1/eps)，可以得到：

ans =

4503599627370496.00000

执行函数 sprintf('%d',round(pi))，可以得到：

ans =

3

执行函数 sprintf('%s','hello')，可以得到：

ans =

hello

执行函数 sprintf('The array is %dx%d.',2,3)，可以得到：

ans =

The array is 2x3.

执行函数 sprintf('\n')，得到一个新行。

调用函数[s,errrmsg]=sprintf('%f','hello')，则可得：

s =

104.000000101.000000108.000000108.000000111.000000

errrmsg =

''

若调用函数[s,errrmsg]=sprintf('%D','hello')，则得到如下所示的错误信息：

s =

Empty string: 1-by-0

errrmsg =

Invalid format.

6. 格式读入字符串

名称：sscanf

格式读入字符串。

语法：该函数有如下几种表达形式：

- A = sscanf(s,format)
- A = sscanf(s,format,size)
- [A,count,errmsg,nextindex] = sscanf(...)

描述：A = sscanf(s,format)从 MATLAB 字符串变量 s 中读入数据，按照字符串 format 指定的格式对该数据进行转换，并将结果返回到矩阵 A 中。

A = sscanf(s,format,size)从 MATLAB 字符串变量 s 中读入 size 个数据，按照字符串 format 指定的格式对该数据进行转换，并将结果返回到矩阵 A 中。参数 size 可以取以下的数值：

取 n，表示往一个列向量中读入 n 个元素。

取 inf，表示读到数据文件尾。

取[m,n]，表示读满一个 m×n 阶矩阵。

[A,count,errmsg,nextindex] = sscanf(...)从 MATLAB 字符串变量 s 中读入 size 个数据，按照字符串 format 指定的格式对该数据进行转换，并将结果返回到矩阵 A 中。参数 count 是一个可选项，它返回已经读入的单元数。参量 errmsg 也是一个可选项，当数据读入有错误时，在 errmsg 中返回一个错误信息；如果没有错误，则返回一个空矩阵。参数 nextindex 返回 s 中已经被读入的字符数加 1 后得到的数。

举例：

首先利用函数 s = '2.7183 3.1416 5.8934 8.1235 3.4345 6.2354 9.1201 4.2352 6.1246'定义字符串，然后调用函数 A = sscanf(s,'%f')，可得：

```
A =
    2.7183
    3.1416
    5.8934
    8.1235
    3.4345
    6.2354
    9.1201
    4.2352
    6.1246
```

若调用函数 A = sscanf(s,'%f',4)，则可得：

```
A =
    2.7183
    3.1416
    5.8934
    8.1235
```

若调用函数[A,count,errmsg,nextindex] = sscanf(s,'%f',4)，则可得到：

```
A =
    2.7183
    3.1416
    5.8934
    8.1235
count =
     4
errmsg =
    ''
nextindex =
    29
```

7. 字符串转换为双精度数

名称：str2double

将字符串转换为双精度数。

语法：该函数有如下两种表达形式：

- x = str2double('str')
- X = str2double(C)

描述：x = str2double('str')将字符串 str 中的元素转换为双精度数。

X = str2double(C)将字符串细胞数组 C 中的字符转换为双精度数。

举例：

例如执行函数 str2double('123.45e7')，可以得到：

```
ans =
  1.2345e+009
```

执行函数 str2double('123 + 45i')，可以得到：

```
ans =
  1.2300e+002 +4.5000e+001i
```

执行函数 str2double('3.14159')，可以得到：

```
ans =
     3.1416
```

执行函数 str2double('2.7i - 3.14')，可以得到：

```
ans =
  -3.1400 + 2.7000i
```

执行函数 str2double('1,200.34')，可以得到：

```
ans =
     1.0000    200.3400
```

8. 字符串转换为数值

名称：str2num

将字符串转换为数值。

语法：x = str2num('str')

描述：x = str2num('str')将字符串 str 中的元素转换为数值。字符串中可以包括阿拉伯数字、小数点、正负号等。

函数 str2num 也可以将字符串矩阵中的元素转换为数值。

举例：

例如执行函数 str2num('3.14159e0')，可以得到：

```
ans =
     3.1416
```

若执行函数 str2num(['1 2';'3 4'])，可以得到：

```
ans =
     1     2
     3     4
```

12.4 基转换

1. 二进制转换为十进制

名称：bin2dec

把二进制转换为十进制。

语法：bin2dec(binarystr)

描述：bin2dec(binarystr)返回二进制字符串 binarystr 对应的十进制数值。

举例：

例如执行函数 bin2dec('010111')，可以得到：

```
ans =
    23
```

2. 十进制转换为二进制

名称：dec2bin

把十进制转换为二进制。

语法：该函数有如下两种表达形式：

- str = dec2bin(d)
- str = dec2bin(d,n)

描述：str = dec2bin(d)返回 d 的二进制字符串 str。其中 d 必须是一个小于 2^{52} 的非负整数。

str = dec2bin(d,n)返回一个至少包含 n 位的二进制字符串。

举例：

例如执行函数 str = dec2bin(34)，可以得到：

```
str =
100010
```

若执行函数 str = dec2bin(34,10)，则可以得到：

```
str =
0000100010
```

3. 十进制转换为十六进制

名称：dec2hex

把十进制转换为十六进制。

语法：该函数有如下两种表达形式：

- str = dec2hex(d)
- str = dec2hex(d,n)

描述：str = dec2hex(d)把十进制整数 d 转换为十六进制字符串 str。其中 d 必须是一个小于 2^{52} 的非负整数。

str = dec2hex(d,n)返回一个至少包含 n 位的十六进制字符串。

举例：

例如执行函数 str = dec2hex (666)，可以得到：

str =

29A

若执行函数 str = dec2hex (666,8)，则可得：

str =

0000029A

4. 十六进制转换为十进制

名称：hex2dec

把十六进制转换为十进制。

语法：d = hex2dec('hex_value')

描述：d = hex2dec('hex_value')将十六进制整数 hex_value 转换为浮点整数表达式。

举例：

例如执行函数 hex2dec('3ff')，可以得到：

ans =

    1023

5. 十六进制转换为双精度数

名称：hex2num

把十六进制转换为双精度数。

语法：f = hex2num('hex_value')

描述：f = hex2num('hex_value')将十六进制数 hex_value 转换为对应的双精度浮点数。

举例：

例如执行函数 f = hex2num('400921fb54442d18')，可以得到：

f =

    3.14159265358979

第十三章　低层文件输入输出函数

13.1　打开和关闭文件

1. 关闭文件

名称：fclose

关闭一个或多个打开的文件。

语法：有如下两种表达形式：

- status = fclose(fid)
- status = fclose('all')

描述：status = fclose(fid)关闭指定的文件，并对操作结果返回一个值。该值见表 13-1。

表 13-1　　status 的值

值	说明
0	关闭指定文件的操作成功
-1	关闭指定文件的操作不成功

status = fclose('all')命令将关闭所有打开的文件。

当使用 status = fclose('all')命令关闭所有打开的文件时，基本的输入、输出及错误处理文件将不被关闭。

2. 打开文件

名称：fopen

打开一个文件或者获得打开文件的信息。

语法：有如下几种表达形式：

- fid = fopen(filename,permission)
- [fid,message] = fopen(filename,permission,format)
- fids = fopen('all')
- [filename,permission, format] = fopen(fid)

描述：fid = fopen(filename,permission)命令将使用指定的模式打开指定的文件，并返回文件的标识符。"permission" 指定的打开模式见表 13-2。

[fid,message]=fopen(filename,permission,format)使用指定的模式打开指定的文件，并返回文件的标识符和信息，此外，用户可以使用"format"指定数字格式。"format"的值见表 13-3。

fids = fopen('all')命令将返回一个包含所有打开文件的标识符的行向量。

[filename,permission, format] = fopen(fid)命令将返回指定文件的全文件名、模式和格式。如果指定的文件名是无效的，该命令将返回一个全为空的行向量。

表 13-2 "permission"指定的打开模式

值	说明
r	只读模式
r+	读写模式
w	删除已存在文件的内容或创建一个新文件，并在写模式下打开该文件
w+	删除已存在文件的内容或创建一个新文件，并在读写模式下打开该文件
W	无自动刷新的写模式
a	写模式下创建并打开一个新文件或打开一个已存在文件并添加到指定文件的末尾
a+	读写模式下创建并打开一个新文件或打开一个已存在文件并添加到指定文件的末尾
A	无自动刷新的添加模式

表 13-3 "format"的值

值	说明
'cray'或'c'	CRAY 浮点格式，big-endian 字节顺序
'ieee-be'或'b'	IEEE 浮点格式，big-endian 字节顺序
'ieee-le'或'l'	IEEE 浮点格式，little-endian 字节顺序
'ieee-be.l64'或's'	IEEE 浮点格式，big-endian 字节顺序，64 位数据类型
'ieee-le.l64'或'a'	IEEE 浮点格式，little-endian 字节顺序，64 位数据类型
'native'或'n'	本机格式，系统缺省值
'vaxd'或'd'	VAX D 浮点格式，VAX 字节顺序
'vaxg'或'g'	VAX G 浮点格式，VAX 字节顺序

当用户使用 fopen 命令时，在缺省状态下，文件将以二进制模式打开，但在有些系统中，文件将以文本模式打开。在 PC 和 VMS 系统中，用户必须区分二进制模式和文本模式；但在 UNIX 系统中，用户不必区分二进制模式和文本模式。

13.2 无格式输入输出

1. 读二进制数据

名称：fread

从文件中读二进制数据。

语法：有如下两种表达形式：

- [A,count] = fread(fid,size,precision)
- [A,count] = fread(fid,size,precision,skip)

描述：[A,count] = fread(fid,size,precision)命令将从指定文件中读取二进制数据并写入矩阵 A 中。可选的输出参数"count"返回元素成功被读取的次数。

可选参数"size"将决定多少数据被读取。如果该参数没有被指定，fread 命令将一直读取到指定文件的结尾。该参数的有效选项见表 13-4。

表 13-4 参数"size"的有效选项

选项	说明
n	读取 n 个元素
inf	读到指定文件的结尾
[m,n]	读取 m×n 个元素

如果 fread 已经读到指定文件的结尾，且当前的输入流没有包含足够的二进制数字去写出一个完整的矩阵，该命令将使用 0 填充空位。

[A,count] = fread(fid,size,precision,skip)命令将从指定文件中读取二进制数据并写入矩阵 A 中。可选的输出参数“count”返回元素成功被读取的次数，可选参数“skip”用于指定每个精度值被读取后跳过的字节数。

precision 控制每个读取值的精度位数，以及把这些数据转换成字符、整数或浮点数。表 13-5 中的字符串，无论是MATLAB 型还是相应的C 或 FORTRAN 型，都可以作为控制 precision 而使用。如果没有特别指定，其缺省值为'uchar'。

表 13-5　　作为控制 precision 的字符串

MATLAB	C 或 FORTRAN	说明
'schar'	'signed char'	8 位带符号字符
'uchar'	'unsigned char'	8 位无符号字符
'int8'	'integer*1'	8 位整数
'int16'	'integer*2'	16 位整数
'int32'	'integer*4'	32 位整数
'int64'	'integer*8'	64 位整数
'uint8'	'integer*1'	8 位无符号整数
'uint16'	'integer*2'	16 位无符号整数
'uint32'	'integer*4'	32 位无符号整数
'uint64'	'integer*8'	64 位无符号整数
'float32'	'real*4'	32 位浮点数
'float64'	'real*8'	64 位浮点数
'double'	'real*8'	64 位浮点数

如果用户不关心可移植性，那么表 13-6 中的数值精度格式也许是更为有用的。

表 13-6　　数值精度格式

MATLAB	C 或 FORTRAN	说明
'char'	'char*1'	8 位字符
'short'	'short'	16 位整数
'int'	'int'	32 位整数
'long'	'long'	32 位或 64 位整数
'ushort'	'unsigned short'	16 位无符号整数
'uint'	'unsigned int'	32 位无符号整数
'ulong'	'unsignde long'	32 位或 64 位无符号整数
'float'	'float'	32 位浮点数

2. 把二进制数据写入文件

名称：fwrite

把二进制数据写入文件。

语法：有如下两种表达形式：

- count = fwrite(fid,A,precision)
- count = fwrite(fid,A,precision,skip)

描述：count = fwrite(fid,A,precision)命令将把矩阵 A 中的所有元素写入到指定文件中，并把 MATLAB 值转换成指定数值精度。

数据将被按列顺序写入到文件中，count 中将记录被成功写入的元素的数目。参数 fid

是文件标识符。

count = fwrite(fid,A,precision,skip)命令将把矩阵 A 中的所有元素写入到指定文件中，并把 MATLAB 值转换成指定数值精度。可选参数“skip”用于指定每个精度值被写入前跳过的字节数。关于数值精度请参见表 13-5 和表 13-6。

13.3　格式输入输出

1. 按行从文件读取数据

名称：fgetl

按行从文件读取数据并放弃换行符。

语法：line = fgetl(fid)

描述：line = fgetl(fid)命令将返回标识符为 fid 的文件的下一行。如果使用该命令时遇到文件的结尾，将返回-1。返回的行将不包括换行符。

2. 从文件中读取行

名称：fgets

从文件中读取行，保留换行符并把行作为字符串返回。

语法：有如下两种表达形式：

- line = fgets(fid)
- line = fgets(fid,nchar)

描述：line = fgets(fid)命令将把标识符为 fid 的文件的下一行作为字符串返回。如果使用该命令时遇到文件的结尾，将返回-1。

line = fgets(fid,nchar)命令将把标识符为 fid 的文件的下--行作为字符串返回，并至多返回用 nchar 指定的字符个数。返回的行将包括换行符。

3. 把格式化数据写入文件

名称：fprintf

把格式化数据写入文件。

语法：有如下两种表达形式：

- count = fprintf(fid,format,A,...)
- fprintf(format,A,...)

描述：count = fprintf(fid,format,A,...)按 FORMAT 指定的格式格式化矩阵 A 的实部和其他附加矩阵变量中的数据，且写入到标识符为 fid 的文件，count 中将记录被成功写入的字节数。

fprintf(format,A,...)命令按 FORMAT 指定的格式格式化矩阵 A 和其他附加矩阵变量中的数据，且写入到标准输出——显示屏。

fprintf 函数和 ANSI C 中的 fprintf()函数类似，具有一些例外和扩展，主要包括：

(1) 由转换符 o、u、x、X 支持的非标准辅助区分符：

b 是一个基本的 C 数据类型，该数据类型与其说是一个无符号整数，到不如说是一个双精度数。

t 是一个基本的 C 数据类型，该数据类型与其说是一个无符号整数，到不如说是一个浮点数。

(2) 当输入矩阵 A 是一个非数字矩阵时，fprintf 被向量化。

格式化字符串是通过矩阵 A 的元素循环，直到这些元素用完。以同样的方式，不需重新启动，也可通过任何附加矩阵向量进行循环。表 13-7 列出了一些非字母数字字符。

表 13-7　　非字母数字字符

值	作用
\b	退格
\f	换页
\n	新行
\r	回车
\t	水平制表
\\	反斜杠
\''	单引号
%%	百分号

举例：

在 MATLAB 命令窗口中输入：

fprintf('Hello,World!')

结果为：

Hello,World!

在 MATLAB 命令窗口中输入：

A = [32.14 86.68;3214 8668];

fprintf(1,'X is %6.2f meters or %8.3f cm\n',A)

结果为：

X is　32.14 meters or 3214.000 cm

X is　86.68 meters or 8668.000 cm

4. 从文件中读格式化数据

名称： fscanf

从文件中读格式化数据。

语法： 有如下两种表达形式：

- A = fscanf(fid,format)
- [A,count] = fscanf(fid,format,size)

描述： A = fscanf(fid,format)命令将从指定标识符为 fid 的文件中读取所有数据，并根据 format 指定的格式对之进行转换，返回道矩阵 A 中。

[A,count] = fscanf(fid,format,size)命令将从指定标识符为 fid 的文件中读取所有数据，并根据 format 指定的格式对之进行转换，返回到矩阵 A 中，可选项 size 用于限制从文件中读取的元素数目，如果没有进行指定，将认为是整个文件。

size 的有效选项见表 13-8。

表 13-8 size 的有效选项

值	说明
n	读取 n 个元素
inf	读到文件结尾
[m,n]	读取足够的元素返回到一个 mn 矩阵中，其中 n 可以为 inf，但 m 不行

13.4 文件定位

1. 文件是否结束

名称：feof

测试文件是否结束。

语法：eofstat = feof(fid)

描述：eofstat = feof(fid)测试指定标识符为 fid 的文件是否设置结束标志。如果该文件设置结束标志，命令将返回 1；否则将返回 0。当不需更多输入时，应设置文件的结束标志。

2. 查询文件错误信息

名称：ferror

查询文件输入、输出错误信息。

语法：有如下几种表达形式：

- message = ferror(fid)
- message = ferror(fid,'clear')
- [message,errnum] = ferror(...)

描述：message = ferror(fid)命令将返回标识符为 fid 的文件的输入、输出错误信息。

message = ferror(fid,'clear')命令将清除标识符为 fid 的文件的输入、输出错误信息。

[message,errnum] = ferror(...)命令将返回标识符为 fid 的文件的输入、输出错误信息，并返回最近的错误状态数目。如果指定文件最近的输入输出操作是成功的，[message,errnum] = ferror(...)命令将返回一个值为空的 message 和值为 0 的 errnum。

3. 反绕一个打开的文件

名称：frewind

反绕一个打开的文件。

语法：frewind(fid)

描述：frewind(fid)命令将把文件位置指针反绕到标识符为 fid 的文件的开头。若反绕一个磁带设备上的文件，可能不进行任何操作。

4. 设置文件位置指针

名称：fseek

设置文件位置指针。

语法：status = fseek(fid,offset,origin)

描述：status = fseek(fid,offset,origin)命令将重新定位代号为 fid 的文件的位置指针，并把文件指针从 origin 指定处(当前指针位置)呆时移动到偏移量为 offset 的位置。

5. 文件位置指针

名称：ftell

获取文件位置指针。

语法：position = ftell(fid)

描述：position = ftell(fid)命令将返回标识符为 fid 的文件的位置指针的位置。返回的位置值表示从文件开头到指针处的字节数。如果返回值为-1，表示查询失败，此时应使用 ferror 命令确定错误类型。

13.5 字符串转换

1. 格式化数据写入到一个字符串

名称：sprintf

把格式化数据写入到一个字符串。

语法：有如下两种表达形式：

- s = sprintf(format,A,...)
- [s,errrmsg] = sprintf(format,A,...)

描述：s = sprintf(format,A,...)命令按 FORMAT 指定的格式格式化矩阵 A 和其他附加矩阵变量中的数据，并返回到 MATLAB 字符串变量 s 中。

[s,errrmsg]=sprintf(format,A,...)按 FORMAT 指定的格式格式化矩阵 A 和其他附加矩阵变量中的数据，并返回到字符串变量 s 中，如果产生错误，将把错误信息返回到 errrmsg 中。

举例：

在 MATLAB 命令窗口中输入：

sprintf('%15.5f',1/eps)

结果为：

ans =

4503599627370496.00000

2. 使用格式控制读取字符串

名称：sscanf

使用格式控制读取字符串。

语法：有如下几种表达形式：

- A = sscanf(s,format)
- A = sscanf(s,format,size)
- [A,count,errmsg,nextindex] = sscanf(...)

描述：A = sscanf(s,format)命令将从 MATLAB 字符变量 s 中读取数据，并转换成指定的格式返回到矩阵 A 中。

A = sscanf(s,format,size)命令将从 MATLAB 字符变量 s 中读取数据，并转换成指定的格式返回到矩阵 A 中，可选项 size 用于限制从文件中读取的元素数目，如果没有进行指定，将认为是整个文件。

[A,count,errmsg,nextindex] = sscanf(...)命令将从 MATLAB 字符变量 s 中读取数据，并转换成指定的格式返回到矩阵 A 中。可选输出参数 count 返回成功被读取的元素的数目；可选输出参数 errmsg 返回错误信息。

举例：

在 MATLAB 命令窗口中输入：

```
s = '9.1897   5.2658';
A = sscanf(s,'%f')
```

结果为：

```
A =
   9.18970000000000
   5.26580000000000
```

13.6　特殊文件输入输出

1. 把 ASCII 限定文件读入到矩阵

名称： dlmread

把一个 ASCII 限定文件读入到矩阵中。

语法： 有如下几种表达形式：

- M = dlmread(filename,delimiter)
- M = dlmread(filename,delimiter,r,c)
- M = dlmread(filename,delimiter,range)

描述： M = dlmread(filename,delimiter)命令将从指定 ASCII 文件中使用 delimiter 指定的分隔符读取数据。逗号是默认的分隔符，使用'\t'指定制表分隔符。

M = dlmread(filename,delimiter,r,c)命令将从指定 ASCII 文件中使用 delimiter 指定的分隔符读取数据，文件的起始偏移量为 r 和 c，其中 r 为行偏移量，c 为列偏移量。逗号是默认的分隔符，使用'\t'指定制表分隔符。

M = dlmread(filename,delimiter,range)命令将从指定 ASCII 文件中使用 delimiter 指定的分隔符读取数据，并引入了一个指定的 ASCII 限定数据的范围。逗号是默认的分隔符，使用'\t'指定制表分隔符。

可以使用如下格式指定 range：

range = [UpperLeftRow UpperLeftColumn LowerRightRow LowerRightColumn]

dlmread 命令将使用 0 来填充空限定域。

2. 把矩阵写入到 ASCII 限定文件

名称： dlmwrite

把一个矩阵写入到 ASCII 限定文件中。

语法： 有如下两种表达形式：

- dlmwrite(filename,A,delimiter)
- dlmwrite(filename,A,delimiter,r,c)

描述：dlmwrite(filename,A,delimiter)命令将把矩阵 A 转换到一个 ASCII 格式文件中，并使用 delimiter 指定的分隔符隔开矩阵中的元素，filename 是数据被写入的电子表单名。逗号是默认的分隔符，使用'\t'产生制表符分隔的文件。

dlmwrite(filename,A,delimiter,r,c)命令将把矩阵 A 转换到一个 ASCII 格式文件中，并使用 delimiter 指定的分隔符隔开矩阵中的元素，filename 是数据被写入的电子表单名。电子表单的起始偏移量为 r 和 c，其中 r 为行偏移量，c 为列偏移量。逗号是默认的分隔符，使用'\t'产生制表符分隔的文件。任何为 0 的元素将被忽略。

举例：

在 MATLAB 命令窗口中输入：

```
dlmwrite('myfile',[1 0 1;0 1 0;1 1 1],',')
type myfile
```

结果为：

```
1,,1
,1,
1,1,1
```

3. HDF 接口

名称：hdf

HDF 接口。

语法：hdf*(functstr,param1,param2,...)

描述：MATLAB 提供了一个函数集合，可供用户访问 HDF 库(一个由 NCSA 开发和支持的函数库)。表 13-9 列出了 MATLAB 中的 HDF 接口。

表 13-9　　MATLAB 中的 HDF 接口

函数	接口
hdfan	多文件注释
hdfdf24	24 位光栅图像
hdfdfr8	8 位光栅图像
Hdfgd	HDF-EOS GD 接口
Hdfh	HDF H 接口
Hdfhd	HDF HD 接口
Hdfhe	HDF HE 接口
Hdfml	通路程序
Hdfpt	HDF-EOS PT 接口
Hdfsd	多文件科学数据集
Hdfsw	HDF-EOS SW 接口
Hdfv	V 组
Hdfvf	V 数据 VF 函数
Hdfvh	V 数据 VH 函数
Hdfvs	V 数据 VS 函数

4. 图形文件的相关信息

名称：imfinfo

返回一个图形文件的相关信息。

语法：有如下几种表达形式：

- info = imfinfo(filename,fmt)
- info = imfinfo(filename)

描述：info = imfinfo(filename,fmt)命令将返回一个结构，其字段包含图形文件中图像的相关信息。filename 是指定图形文件的文件名，frm 是一个指定格式的字符串。参数 frm 的可选项见表 13-10。

表 13-10　　　　参数 frm 的可选项

format	文件类型
'bmp'	BMP 格式
'hdf'	HDF 格式
'jpg' or 'jpeg'	JPG 格式
'pcx'	PCX 格式
'png'	PNG 格式
'tif' or 'tiff'	TIFF 格式
'xwd'	XWD 格式

info = imfinfo(filename)命令将返回一个结构，其字段包含图形文件中图像的相关信息。

使用该命令时，指定的图形文件必须在 MATLAB 的当前目录或 MATLAB 路径目录中。

举例：

在 MATLAB 命令窗口中输入 info = imfinfo('myfig.jpg')，

将显示：

```
info =          Filename: 'myfig.jpg'
             FileModDate: '19-Oct-1999 21:15:22'
                FileSize: 17727
                  Format: 'jpg'
           FormatVersion: ''
                   Width: 1200
                  Height: 900
                BitDepth: 24
               ColorType: 'truecolor'
         FormatSignature: ''
```

5. 读取图像

名称：imread

从图形文件中读取图像。

语法：有如下几种表达形式：

- A = imread(filename,fmt)
- [X,map] = imread(filename,fmt)
- [...] = imread(filename)
- [...] = imread(...,idx)

描述：A = imread(filename,fmt)命令将从名为 filename 的图形文件中读取灰度或真彩图到 A 中。如果文件中的图像为灰度图，A 是一个两维数组；如果文件中的图像为真彩图，A 是一个三维数组。

[X,map] = imread(filename,fmt)命令将从名为 filename 的图形文件中读取图像到 X 中，并将其色彩映射返回到 map 中。

[...] = imread(filename)命令将根据图形文件的内容来推断其格式。

[...] = imread(...,idx)命令将从多图像 TIFF 文件中读取一个图像，idx 是一个整数值，该值用于指定该图像在图形文件中的显示次序。

6. 图像写入到图形文件

名称：imwrite

把图像写入到一个图形文件中。

语法：有如下几种表达形式：

- imwrite(A,filename,fmt)
- imwrite(X,map,filename,fmt)
- imwrite(...,filename)
- imwrite(...,Param1,Val1,Param2,Val2...)

描述：imwrite(A,filename,fmt)命令将把 A 中的图像写入到名为 filename 的指定文件中。fmt 是指定文件格式的字符串。

imwrite(X,map,filename,fmt)命令将把 A 中的图像和其色彩映射 map 写入到名为 filename 的图形文件中，fmt 是指定文件格式的字符串。

imwrite(...,filename)命令将把图像写入到名为 filename 的指定文件中，并根据指定文件的扩展名来推断图像的格式。此时，指定文件的扩展名必须为 fmt 的合法值之一。

imwrite(...,Param1,Val1,Param2,Val2...)命令将指定参数来控制输出文件的不同特征。

HDF 格式文件的有效参数见表 13-11。

表 13-11　　HDF 格式文件的有效参数

参数	值
'Compression'	值为'none'、'rle'、'jpeg'其中之一；'rle'仅对色彩指数图像有效，'jpeg'仅对 RGB 图像有效
'Quality'	值为 0 到 100 间的一个数字；该参数仅应用于参数'Compression'的值为'jpeg'时
'WriteMode'	值为'overwrite'和'append'其中之一

JPEG 格式文件的有效参数见表 13-12。

表 13-12　　JPEG 格式文件的有效参数

参数	值
'Quality'	值为 0 到 100 间的一个数字；越大的数字意味越好的质量，但同时意味着文件将越大

TIFF 格式文件的有效参数见表 13-13。

表 13-13　　TIFF 格式文件的有效参数

参数	值
'Compression'	值为'none'、'packbits'、'ccitt'其中之一；'ccitt'仅对二进制图像有效，'packbits'仅对非二进制图像有效
'Description'	任意字符串
'Resolution'	一个用于在 x 和 y 方向设置输出文件的分辨率

PNG 格式文件的有效参数见表 13-14。

表 13-14　　　　　　　　　PNG 格式文件的有效参数

参数	值
'Author'	一个字符串
'Description'	一个字符串
'Copyright'	一个字符串
'CreationTime'	一个字符串
'Software'	一个字符串
'Disclaimer'	一个字符串
'Warning'	一个字符串
'Source'	一个字符串
'Comment'	一个字符串
'InterlaceType'	'none'和'adam7'其中之一
'BitDepth'	一个指示位深度的数值
'Transparency'	仅当非 alpha 通道被使用时用于指示透明度信息
'Background'	当合成透明像素时被用于指定背景色的数值
'Gamma'	一个非负数值
'Chromaticities'	一个向量，带有 8 个元素[wx wy rx ry gx gy bx by]，用于指定色度
'XResolution'	一个用于说明水平方向的像素数目的数值
'YResolution'	一个用于说明垂直方向的像素数目的数值
'ResolutionUnit'	'unknown'和'meter'其中之一
'Alpha'	一个为每个像素指定透明度的矩阵
'SignificantBits'	一个用于说明数据数组中有效位的多少的数值或向量，其值必须在区间[1,bitdepth]内

表 13-15 列出了 imwrite 命令能够写入的图像的类型摘要。

表 13-15　　　　　　　imwrite 命令能够写入的图像的类型摘要

格式	差异
BMP	带色彩映射的 8 位未压缩图像；24 位未压缩图像
HDF	8 位光栅图像数据集，带或不带色彩映射；24 位光栅图像数据集
JPEG	基准 8 位或 24 位 JPEG 图像
PCX	8 位图像
TIFF	基准 TIFF 图像，包括 1 位、8 位和 24 位未压缩图像；1 位、8 位和 24 位按位压缩图像；1 位 CCITT 压缩图像
XWD	8 位 ZPixmaps

7. 文本文件中读取格式化数据

名称：textread

从文本文件中读取格式化数据。

语法：有如下两种表达形式：

- [A,B,C,...] = textread('filename','format')
- [A,B,C,...] = textread('filename','format',N)

描述：[A,B,C,...] = textread('filename','format')命令将把数据从指定文本文件中使用指定的格式读到 A，B，C 等变量中。使用 filename 指定文件名，使用 format 指定格式。

字符串 format 决定了返回参数的数目和类型。format 的有效取值见表 13-16。

表 13-16 format 的有效取值

格式	作用
普通字符	忽略匹配的字符
%d	读入一个单精度整数值
%u	读入一个整数值
%f	读入一个浮点值
%s	读入一个空格分隔的字符串
%q	读入一个字符串
%c	读入包括空格的字符
%[...]	读入用方括号内字符指定的最长字符串
%[^...]	读入未用方括号内字符指定的最长非空字符串
%*...	忽略由*指定的匹配字符
%w...	读入由 w 指定宽度的字段

[A,B,C,...] = textread('filename','format',N)命令将把数据从指定文本文件中使用指定的格式读到 A，B，C 等变量中，并重复操作 N 次。这里 N 为正整数，当 N 小于 0 时，该命令将读整个指定文件。

8. 把 Lotus123 WK1 电子表格读入到矩阵

名称：wk1read

把一个 Lotus123 WK1 电子表格读入到一个矩阵中。

语法：有如下几种表达形式：

- M = wk1read(filename)
- M = wk1read(filename,r,c)
- M = wk1read(filename,r,c,range)

描述：M = wk1read(filename)命令将把一个 Lotus123 WK1 电子表格读入到矩阵 M 中。Filename 是指定的电子表格名。

M = wk1read(filename,r,c)命令将从一个 Lotus123 WK1 电子表格的指定位置把数据读入到矩阵 M 中。Filename 是指定的电子表格名，r 和 c 分别是行和列的偏移量。

M = wk1read(filename,r,c,range)命令将从一个 Lotus123 WK1 电子表格的指定位置把数据读入到矩阵 M 中。Filename 是指定的电子表格名，r 和 c 分别是行和列的偏移量，参数 range 规定了读入数据值的范围。

9. 把矩阵中的数据写入到 Lotus123 WK1 电子表格

名称：wk1write

把一个矩阵中的数据写入到一个 Lotus123 WK1 电子表格中。

语法：有如下两种表达形式：

- wk1write(filename,M)
- wk1write(filename,M,r,c)

描述：wk1write(filename,M)命令将把矩阵 M 中的数据写入到指定的 Lotus123 WK1 电子表格中。Filename 是指定的电子表格名。

wk1write(filename,M,r,c)命令将把矩阵 M 中的数据按照指定的位置写入到指定的 Lotus123 WK1 电子表格中。Filename 是指定的电子表格名，r 和 c 分别是行和列的偏移量。

第十四章　位操作、结构和对象函数

14.1　位操作函数

1.　按位与

名称：bitand

按位与。

语法：C = bitand(A,B)

描述：C = bitand(A,B)命令将返回两个非负整数数组 A 和 B 的相应元素进行按位与操作的结果。为了确保 A 和 B 中的元素都是整数，可以使用 ceil、fix、floor、和 round 函数来产生 A 和 B。

举例：

在 MATLAB 命令窗口中输入：

```
A = [2 7 37;15 24 35;77 5 28;0 11 10];
B = [1 2 3;4 5 6;7 8 9;10 11 12];
C = bitand(A,B)
```

结果为：

```
C =
     0     2     1
     4     0     2
     5     0     8
     0    11     8
```

2.　按位求补

名称：bitcmp

按位求补。

语法：C = bitcmp(A,n)

描述：C = bitcmp(A,n)命令将把 A 中相应元素的补数作为 n 位浮点整数返回到 C 中。A 中元素的值应为小于等于 2^n 的非负整数。

举例：

在 MATLAB 命令窗口中输入：

```
A = [255 0 37;15 254 1;177 125 28;223 11 128];
C = bitcmp(A,8)
```

结果为：

```
C =
```

```
     0   255   218
   240     1   254
    78   130   227
    32   244   127
```

3. 按位或

名称：bitor

按位或。

语法：C = bitor(A,B)

描述：C = bitor(A,B)命令将返回两个非负整数数组 A 和 B 的相应元素进行按位或操作的结果。为了确保 A 和 B 中的元素都是整数，可以使用 ceil、fix、floor、和 round 函数来产生 A 和 B。

举例：

在 MATLAB 命令窗口中输入 C = bitor(13,186)，

结果为：

```
C =
   191
```

继续输入：

```
A = [255 0 37;15 254 1;177 125 28;223 11 128];
B = fix(100*rand(4,3))
B =
    84    83    83
    52     1    50
    20    68    70
    67    37    42
C = bitor(A,B)
```

结果为：

```
C =
   255    83   119
    63   255    51
   181   125    94
   223    47   170
```

4. 最大浮点整数

名称：bitmax

最大浮点整数。

语法：bitmax

描述：bitmax 返回计算机系统的最大无符号浮点整数。在 IEEE 计算机上，该值为 2^53-1。

举例：

在 MATLAB 命令窗口中输入 bitmax，

结果为：

ans =

  9.0072e+015

为了验证用户计算机系统，继续输入 2^53-1

结果为：

ans =

  9.0072e+015

5.　设置指定位的值

名称：bitset

设置指定位的值。

语法：有如下两种表达形式：

- C = bitset(A,bit)
- C = bitset(A,bit,v)

描述：C = bitset(A,bit)命令将把 A 中元素的第 bit 位设为 1。

C = bitset(A,bit,v)命令将把 A 中元素的第 bit 位设为 v，其中，v 必须为 0 或 1。A 中的元素必须为非负整数，bit 必须为 1 到 A 中元素浮点整数表示法的位数之间的一个数字。

举例：

在 MATLAB 命令窗口中输入：

A = [1 2 3;4 5 6;7 8 9];

C = bitset(A,7)

结果为：

```
C =
    65    66    67
    68    69    70
    71    72    73
```

继续输入 C = bitset(A,7,0)，

结果为：

```
C =
     1     2     3
     4     5     6
     7     8     9
```

6.　逐位移动

名称：bitshift

逐位移动。

语法：有如下两种表达形式：

- C = bitshift(A,k)
- C = bitshift(A,k,n)

描述：C = bitshift(A,k,n)命令将把 A 中元素值移动 k 位后返回到 C 中。n 为规定的浮点整数的位数。

C = bitshift(A,k)命令将把 A 中元素值移动 k 位后返回到 C 中。N 取默认值 53。

如果 k>0，该命令的作用相当于用 2^k 去乘 A 中的元素；如果 k<0，该命令的作用相当于用 2^k 去除 A 中的元素。如果该命令的操作结果超出了规定的浮点整数的位数 n，溢出位将被舍弃。A 中的元素必须为非负整数。

举例：

在 MATLAB 命令窗口中输入：

```
A = [0 1 128;15 81 127;216 58 255];
C = bitshift(A,4)
```

结果为：

```
C =
        0          16        2048
      240        1296        2032
     3456         928        4080
```

继续输入：

```
C = bitshift(A,4,8)
```

将舍弃超出 8 位浮点整数的部分，结果为：

```
C =
     0    16     0
   240    16   240
   128   160   240
```

7. 获取指定位的值

名称：bitget

获取指定位的值。

语法：C = bitget(A,bit)

描述：C = bitget(A,bit)命令将返回 A 中元素用 bit 指定的位的值。A 中的元素必须为非负整数，bit 必须为 1 到 A 中元素浮点整数表示法的位数之间的一个数字。

举例：

在 MATLAB 命令窗口中输入：

```
A = [0 1;128 253;15 81;127 216;58 255];
C = bitget(A,6)
```

结果为：

```
C =
     0     0
     0     1
     0     0
     1     0
     1     1
```

8. 按位异或

名称：bitxor

按位异或。

语法：C = bitxor(A,B)

描述：C=bitxor(A,B)返回两个非负整数数组 A 和 B 的相应元素进行按位异或的结果。为了确保 A 和 B 中的元素都是整数，可使用 ceil、fix、floor、和 round 函数来产生 A 和 B。

举例：

在 MATLAB 命令窗口中输入：

```
C = bitxor(13,255)
```

结果为：

```
C =
   242
```

继续输入：

```
A = [0 1;128 253;15 81;127 216;58 255];
B = [36 2;19 85;0 96;255 1;18 228];
C = bitxor(A,B)
```

结果为：

```
C =
    36     3
   147   168
    15    49
   128   217
    40    27
```

14.2　结构函数

1.　输入处理

名称：deal

把输入处理成输出。

语法：有如下两种表达形式：

- [Y1,Y2,Y3,...] = deal(X)
- [Y1,Y2,Y3,...] = deal(X1,X2,X3,...)

描述：[Y1,Y2,Y3,...] = deal(X)复制 X 到所有的输出 Y1、Y2、Y3……。该命令的作用等于 Y1=X，Y2=X，Y3=X……。

[Y1,Y2,Y3,...] = deal(X1,X2,X3,...)复制 X1、X2、X3……到相应的 Y1、Y2、Y3……。

当使用通过逗号分隔列表扩展单元数组和结构时，deal 命令是极其有用的，下面给出一些常用的句式。

[S.field] = deal(X)语句将把结构 S 中所有名为 field 的字段设置为 X。

[X{:}] = deal(A.field)语句拷贝所有名为 field 的字段中的值到一个单元数组 X 中。

[Y1,Y2,Y3,...] = deal(X{:})语句拷贝单元数组 X 中的值到不同的变量 Y1、Y2、Y3 等中。

[Y1,Y2,Y3,...] = deal(S.field)语句拷贝结构数组 S 中的所有名为 field 的字段中的内容到

不同的变量 Y1、Y2、Y3……。

举例:

在 MATLAB 命令窗口中输入:

```
C = {rand(4) eye(4) ones(4,2) zeros(4,1)};
[Y1,Y2,Y3,Y4] = deal(C{:})
```

结果为:

```
Y1 =
    0.8600    0.8998    0.6602    0.5341
    0.8537    0.8216    0.3420    0.7271
    0.5936    0.6449    0.2897    0.3093
    0.4966    0.8180    0.3412    0.8385
Y2 =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
Y3 =
     1     1
     1     1
     1     1
     1     1
Y4 =
     0
     0
     0
     0
```

2. 字段名

名称: fieldnames

结构的字段名。

语法: names = fieldnames(s)

描述: names = fieldnames(s)命令将返回一个包含结构 s 的字段名的字符串单元数组。

举例:

在 MATLAB 命令窗口中输入:

```
employeeStr(1,1).name = 'liangh';
employeeStr(1,1).age = 25;
employeeStr(1,1).ID = 0001;
employeeStr(1,1).gender = 'man';
```

建立一个结构，继续输入:

```
n = fieldnames(employeeStr)
```

结果为：

n =

'name'

'age'

'ID'

'gender'

3. 字段

名称：getfield

获得结构数组的字段。

语法：有如下两种表达形式：

- f = getfield(s,'field')
- f = getfield(s,{i,j},'field',{k})

描述：f = getfield(s,'field')命令将返回结构 s 中指定字段 field 中的内容到 f 中。该命令的作用相当于 f=s.field。

f = getfield(s,{i,j},'field',{k})命令将返回结构 s 中指定位置的字段 field 中的内容到 f 中。

举例：

在 MATLAB 命令窗口中输入：

```
employeeStr(1,1).name = 'liangh';
employeeStr(1,1).age = 25;
employeeStr(1,1).ID = 0001;
employeeStr(1,1).gender = 'man';
```

建立一个结构，继续输入：

```
f = getfield(employeeStr,{1,1},'ID')
```

结果为：

f = 1

4. 删除结构中的字段

名称：rmfield

删除结构中的字段。

语法：有如下两种表达形式：

- s = rmfield(s,'field')
- s = rmfield(s,FIELDS)

描述：s = rmfield(s,'field')命令将删除结构数组 s 中的名为 field 的字段。

s = rmfield(s,FIELDS)命令将同时删除结构数组 s 中的多个字段。其中，FIELDS 是一个元素为字段名的字符数组。

5. 设置结构数组中的字段

名称：setfield

设置结构数组中的字段。

语法：有如下两种表达形式：

- s = setfield(s,'field',v)

- s = setfield(s,{i,j},'field',{k},v)

描述：s = setfield(s,'field',v)命令将把结构数组 s 中的指定字段 field 的值设置为 v。

s = setfield(s,{i,j},'field',{k},v)命令将把结构数组 s 中的指定位置的字段 field 的值设置为 v。该语句的作用相当于 s(i,j).field(k) = v。

举例：

在 MATLAB 命令窗口中输入：

```
employeeStr(1,1).name = 'liangh';
employeeStr(1,1).age = 25;
employeeStr(1,1).ID = 0001;
employeeStr(1,1).gender = 'man';
```

建立一个结构，继续输入：

```
s = setfield(employeeStr,{1,1},'name','yd')
```

结果为：

```
s =
        name: 'yd'
         age: 25
          ID: 1
      gender: 'man'
```

6. 结构数组

名称：struct

创建一个结构数组。

语法：s = struct('field1',values1,'field2',values2,...)

描述：s=struct('field1',values1,'field2',values2,...)使用指定的字段和值创建一个结构数组。

举例：

在 MATLAB 命令窗口中输入：

```
s = struct('name',{'liangh','yd'},'age',{25 26},'ID',{1},'gender',{'man'})
```

结果为：

```
s =
1x2 struct array with fields:
    name
    age
    ID
    gender
```

7. 结构数组转化为单元数组

名称：struct2cell

把结构数组转化为单元数组。

语法：c = struct2cell(s)

描述：c = struct2cell(s)命令将把带有 p 个字段的 m×n 的结构数组 s 转化为一个 p×m×n 的单元数组 c。如果结构 s 是一个多维结构，则转化后的单元数组 c 为[p size(s)]。

举例：

在 MATLAB 命令窗口中输入：

```
s = struct('name',{'liangh','yd'},'age',{25 26},'ID',{1},'gender',{'man'});
c = struct2cell(s)
```

结果为：

```
c(:,:,1) =
    'liangh'
    [      25]
    [       1]
    'man'
c(:,:,2) =
    'yd'
    [ 26]
    [  1]
    'man'
```

14.3 对象函数

1. 创建对象

名称：class

创建对象或返回对象的类。

语法：有如下几种表达形式：

- str = class(object)
- obj = class(s,'class_name')
- obj = class(s,'class_name',parent1,parent2...)

描述：str = class(object)返回一个规定对象的类的字符串。MATLAB 的类见表 14-1。

表 14-1 MATLAB 中的类

类名	说明
cell	多维单元数组
double	多维双精度数组
sparse	两维稀疏数组
char	字符数组
struct	结构
'class_name'	用户定义类

obj = class(s,'class_name')命令将以结构 s 为模板创建类名为 class_name 的类的对象。

obj = class(s,'class_name',parent1,parent2...)命令将以结构 s 为模板创建类名为 class_name 的类的对象，并保证新创建的对象继承了指定父对象 parent1、parent2 等的方法。

2. 检查对象

名称：isa

检查对象是否属于所给定的类。

语法：K = isa(obj,'class_name')

描述：K = isa(obj,'class_name')命令当所给的对象 obj 属于“class_name”指定的类时，将返回逻辑真(1)；否则将返回逻辑假(0)。

举例：

在 MATLAB 命令窗口中输入：

```
A = rand(3);
isa(A,'cell')
```

结果为 ans = 0。

继续输入：

```
B = ['HELLO THE WORLD!'];
isa(B,'char')
```

结果为 ans = 1。

第十五章　数组函数

15.1　单元数组函数

1. 创建单元数组

名称：cell

创建单元数组。

语法：该函数有如下六种表达形式：

- c=cell(n)
- c = cell(m,n)
- c = cell([m n])
- c = cell(m,n,p,...)
- c = cell([m n p ...])
- c = cell(size(A))

描述：c = cell(n)产生一个 n×n 维的空数组。如果 n 不是标量则会显示错误信息。

c = cell(m,n)或者 c = cell([m,n])产生一个 m×n 维的空数组。要求参数 m 和 n 必须是两个标量。

c = cell(m,n,p,...)或者 c = cell([m n p ...])能产生一个 m×n×p×…维的空数组。同样要求 m，n，p 等参数必须是标量。

c = cell(size(A))能产生一个和所包含的数组 A 阶数完全相同的空数组。

举例：

(1)　A = eye(2)

```
A =
      1      0
      0      1
```

(2)　c = cell(size(A))

```
c =
      []     []
      []     []
```

2. 对单元数组里的每一个元素调用一个函数

名称：cellfun

对单元数组里的每一个元素调用一个函数。

语法：该函数有如下三种表达形式：

- D = cellfun('fname',C)

- D = cellfun('size',C,k)
- D = cellfun('isclass',C,classname)

描述：D = cellfun('fname',C)对数组 C 中的每个元素调用函数 fname 并将结果返回到数组 D 中。D 中的每个元素都是对 C 中相应元素调用函数 fname 后的返回值。输出的数组 D 和数组 C 阶数相同。

所支持的函数如下表：

表 15-1 cellfun 所支持的函数表

函数	返回值
Isempty	对空的数组元素为真
Islogical	对一个逻辑型的数组元素为真
Isreal	对一个实数的数组元素为真
Length	数组元素的长度
Ndims	数组元素的维数
Prodofsize	数组元素中元素的个数

D = cellfun('size',C,k)返回 C 中每个元素第 k 维的维数。

D = cellfun('isclass',C,'classname')对 C 中每个和 classname 相匹配的元素返回真。但这个语句对 classname 子集的对象返回假。

举例：

考虑下面 2×2 的数组：

```
C{1,1} = [1 2; 3 4];
C{1,2} = 'abc';
C{2,1} = 3 + 4i;
C{2,2} = 5;
```

cellfun 返回另外一个 2×2 的数组：

```
D = cellfun('isreal',C)
D =
     1     1
     0     1
len = cellfun('length',C)
len =
     2     3
     1     1
isdbl = cellfun('isclass',C,'double')
isdbl =
     1     0
     1     1
```

3. 从字符数组中创建字符串单元数组

名称：cellstr

从字符数组中创建字符串单元数组。

语法：c = cellstr(S)

描述：c=cellstr(S)将字符数组 S 中的每一行放入 c 中的独立单元。c 可以由一个字符串函数来转换回字符串矩阵。

举例：

给定一个字符串矩阵：

```
S =
Hello
world
```

命令 c = cellstr(S)返回一个 2×1 的数组：

```
c =
    'Hello'
    'world'
```

4. 将单元数组转换为结构数组

名称：cell2struct

将单元数组转换为结构数组。

语法：s = cell2struct(c,fields,dim)

描述：s = cell2struct(c,fields,dim)通过将 c 中 dim 维上的元素调入 s 中的 fields 把数组 c 转换为结构体 s。c 在指定维上的长度必须和 fields 中 fields 名的个数相匹配。要求 fields 必须是一个字符数组或字符串的单元数组。

举例：

```
c = {'LiMing',174,22};
f = {'name','height','age'};
s = cell2struct(c,f,2)
s =
      name: 'LiMing'
    height: 174
       age: 22
```

5. 显示单元数组的内容

名称：celldisp

显示单元数组的内容。

语法：该函数有如下两种表达形式：

- celldisp(C)
- celldisp(C,name)

描述：celldisp(C)递归地显示一个数组的内容。celldisp(C,name)用字符串名 name 代替开始输入的名字（或系统默认的 ans）用于显示。

举例：

用 celldisp 显示一个 2×2 数组的内容：

```
C = {[1 2] 'HY'; 3 'dc'};
celldisp(C)
C{1,1} =
```

```
        1        2
C{2,1} =
        3
C{1,2} =
HY
C{2,2} =
dc
```

6. 图形显示单元数组

名称：cellplot

图形显示单元数组。

语法：该函数有如下三种表达形式：

- cellplot(c)
- cellplot(c,'legend')
- handles = cellplot(...)

描述：cellplot(c)显示一个图形窗口来图示 c 的内容。被填充的矩形代表向量和数组的元素，而标量和短的文本串按原文显示出来。

cellplot(c,'legend')在图形后面接着给出一个图例。

handles = cellplot(c)显示一个图形窗口并返回一个面句柄向量。

举例：

考虑一个 2×2 的数组，它包括一个矩阵、一个向量和两个正文字符串：

```
c{1,1} = '2-by-2';
c{1,2} = 'eigenvalues of eye(2)';
c{2,1} = eye(2);
c{2,2} = eig(eye(2));
```

命令 cellplot(c)得到：

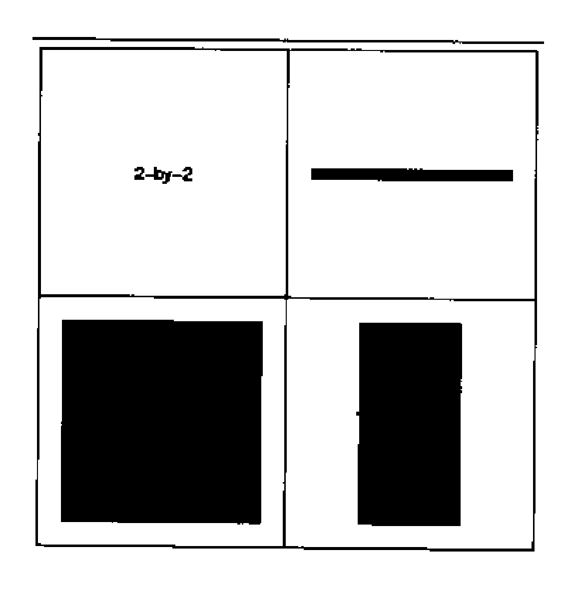


图 15-1　cellplot(c)的结果图

7. 将数值数组转换为单元数组

名称：num2cell

将一个数值数组转换为一个单元数组。

语法：该函数有如下两种表达式：

- c = num2cell(A)
- c = num2cell(A,dims)

描述：c = num2cell(A)通过把 A 中的每个元素放入一个独立的单元从而将矩阵 A 转换为一个数组。数组 c 将和矩阵 A 有相同的阶数。

c = num2cell(A,dims)也将矩阵 A 转换为一个数组，但它是通过把由 dims 指定的维作为一个元素放入相应的单元而实现的。除了和 dims 相应的维数为 1 之外，c 和 A 有相同的阶数。

举例：

表达式

num2cell(A,2)

把 A 的各行放入独立的单元。类似地，

num2cell(A,[1 3])

将 A 的列和深度组成的各页放入独立的单元。

15.2 多维数组函数

1. 连接数组

名称：cat

连接数组。

语法：该函数有如下两种表达式：

- C = cat(dim,A,B)
- C = cat(dim,A1,A2,A3,A4...)

描述：C = cat(dim,A,B)在 dim 维上连接数组 A 和 B。

C = cat(dim,A1,A2,A3,A4,...)在 dim 维上连接所有输入的数组（A1,A2,A3,A4 等等）。

cat(2,A,B)和[A,B]等价，而 cat(1,A,B)和[A;B]相同。

使用由逗号隔开的列表语法结构，cat(dim, C{:})或者 cat(dim, C.field)，可以很方便地将一个包含数字矩阵的单元或结构数组连接为一个单一的矩阵。

举例：

如果给定，

```
A =                    B =
      1    2                 5    6
      3    4                 7    8
```

在不同方向连接可以得到：

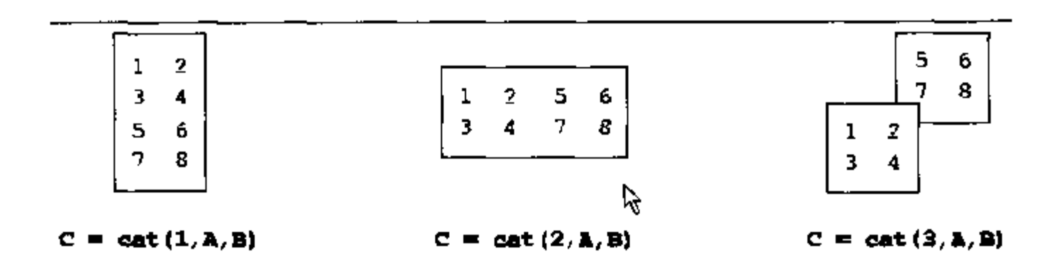


图 15-2　cath 的示例图

命令

A = eye(3);

B = pascal(3);

C = cat(4,A,B);

可以生成一个 3×3×1×2 的数组。

2. 沿指定的方向翻转数组

名称：flipdim

沿指定的方向翻转数组。

语法：B = flipdim(A,dim)

描述：B = flipdim(A,dim)返回数组 A 沿 dim 维翻转后得到的数组。

当 dim 的值为 1 时，数组在行的方向翻转。当 dim 等于 2 时，数组在列的方向翻转。

flipdim(A,1)等价于 flipud(A)，而 flipdim(A,2)则和 fliplr(A)相同。

举例：

当

A =

```
     1     2
     3     4
     5     6
```

flipdim(A,1)得到

```
     5     6
     3     4
     1     2
```

3. 线性索引下标

名称：ind2sub

线性索引下标。

语法：该函数有如下两种表达形式

- [I,J] = ind2sub(siz,IND)
- [I1,I2,I3,...,In] = ind2sub(siz,IND)

描述：命令 ind2sub 确定了和数组的单一索引等价的下标值。

[I,J] = ind2sub(siz,IND)返回数组 I 和 J，I 和 J 中的元素是相应于一个尺寸为 siz 的矩阵的索引矩阵 IND 的行和列的下标。

对于矩阵，[I,J] = ind2sub(size(A)，find(A>5))与[I,J] = find(A>5)返回相同的值。[I1,I2,I3,...,In] = ind2sub(siz,IND)对于一个尺寸为 siz 的数组返回 n 个下标数组 I1，I2，…，In，这些数组中的元素是相应于上述数组的多维索引数组的下标值。

举例：

对于一个 2×2×2 的数组，由和下标等价的线形索引绘制的图如下：

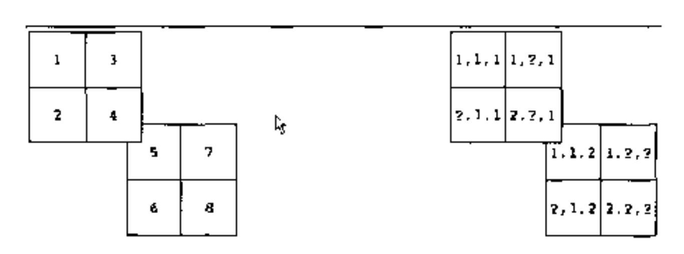


图 15-3　ind2sub 的示例图

4. 多维数组行和列的逆置换

名称： ipermute

多维数组维数的逆置换。

语法： A = ipermute(B,order)

描述： A = ipermute(B,order)是 B 置换的逆。ipermute 重新排列 B 的各维而由 permute(A,order)可重新得到 B。B 和 A 有同样的值，但访问任一特定的元素所需的下标必须按 order 指定的顺序重组。order 中的所有的元素必须各不相同。

permute 和 ipermute 是对多维数组转置(')的一般化。

举例：

考虑一个 2×2×3 的数组 a：

```
a = cat(3,ones(2),2*ones(2),3*ones(2))
a(:,:,1) =                    a(:,:,2) =
        1     1                       2     2
        1     1                       2     2
a(:,:,3) =
        3     3
        3     3
```

以同样的方式对 a 进行序列变换和逆序列变换可以是数组恢复原来的形式。

```
B = permute(a,[3 2 1]);
C = ipermute(B,[3 2 1]);
isequal(a ,C)
ans=
   1
```

5. 生成多维函数和插值数组

名称： ndgrid

生成多维函数和插值数组。

语法： 该函数有如下两种表达形式：

- [X1,X2,X3,...] = ndgrid(x1,x2,x3,...)
- [X1,X2,...] = ndgrid(x)

描述：[X1,X2,X3,...] = ndgrid(x1,x2,x3,...)把由向量 x1，x2，x3 指定的区域转换到数组 X1，X2，X3…中，这可以用于多变量函数的计算和多维添写。第 i 元输出数组是向量 xi 中元素的拷贝。

[X1,X2,...] = ndgrid(x)等价于[X1,X2,...] = ndgrid(x,x,...)。

举例：

在$-2 < x_1 < 2$和$-2 < x_2 < 2$的范围内，计算函数 $x_1*exp(-x_1*x_1-x_2*x_2)$的值。

```
[X1,X2] = ndgrid(-2:.2:2,-2:.2:2);
Z = X1 .* exp(-X1.^2 - X2.^2);
mesh(Z)
```

函数 ndgrid 和 meshgrid 一样，除了输出的前两个量的次序调换了一下。也就是说，表达式

```
[X1,X2,X3] = ndgrid(x1,x2,x3)
```

和下式得到同样的结果。

```
[X2,X1,X3] = meshgrid(x2,x1,x3)
```

因此，ndgrid 比较适合于不是基于空间的多维问题，而 meshgrid 比较适合于二维或三维笛卡尔空间的问题。

6. 数组维数

名称：ndims

数组维数。

语法：n = ndims(A)

描述：n = ndims(A)返回数组 A 的维数。数组的维数总是大于或等于 2。后面的单独维被忽略了。所谓一个单独维是指任何满足 size(A,dim) = 1 的尺度。

7. 重新安排多维数组的行和列

名称：permute

重新安排多维数组的行和列。

语法：B = permute(A,order)

描述：B = permute(A,order)按照向量 order 指定的顺序重新排列 A 的各维。B 和 A 有同样的值，但访问任一特定元素所需的下标需按 order 指定的顺序重新排列。order 的所有元素必须各不相同。

permute 和 ipermute 是一般化的多维数组的转置。

举例：

(1) 对于任意给定的数组 B，表达式

```
permute(B,[2 1])
```

等价于 B 的转置矩阵。

例如：

```
B = [1 1; 2 2];   permute(B,[2 1])
ans =
```

```
        1    2
        1    2
```

(2) 下面的代码重组一个三维的数组：

```
X = [5,6,7];
Y = permute(X,[3 1 2]);
size(Y)
ans =
    7    5    6
```

8. 整形数组

名称：reshape

整形数组。

语法：该函数有如下四种表达形式：

- B = reshape(A,m,n)
- B = reshape(A,m,n,p,...)
- B = reshape(A,[m n p...])
- B = reshape(A,siz)

描述：B = reshape(A,m,n)返回一个 m×n 的矩阵 B，B 中的元素由 A 中按列的顺序取得。如果 A 中元素个数不是 m×n 时将发生错误。

B = reshape(A,m,n,p,...)或者 B = reshape(A,[m n p...])返回一个和 A 有同样元素的数组，但变形后的数组 B 的尺寸为 m×n×p×…。m×n×p×…必须等于 prod(size(x))。

B = reshape(A,siz)返回一个和 A 有相同元素的数组，但变形为向量 siz 所代表地尺寸。数量 prod(siz)必须等于 prod(size(A))。

举例：

把一个 3×4 的矩阵变形为一个 2×6 的矩阵：

```
A =
    1    4    7    10
    2    5    8    11
    3    6    9    12
B = reshape(A,2,6)
B =
    1    3    5    7    9    11
    2    4    6    8    10   12
```

9. 转换行列

名称：shiftdim

转换行列。

语法：该函数有如下两种表达形式：

- B = shiftdim(X,n)
- [B,nshifts] = shiftdim(X)

描述：B = shiftdim(X,n)将 X 的各维移动 n 次。当 n 是正数时，shiftdim 把各维左移并把

开头的 n 个维连接到最末维后面。当 n 是负数时，shiftdim 将各维右移并且每移动一次补上一个单一维。

[B,nshifts] = shiftdim(X)返回一个和 X 有相同数目元素的数组 B，但从开头起到第一个非 1 维的所有单一维都被去掉了。单一维是指任何满足 size(A,dim) = 1 的维。nshifts 是被去掉的维的数目。

如果 X 是一个标量，shiftdim 没有结果。

举例：

shiftdim 命令很容易生成像 sum 和 diff 这些从第一个非单一维开始运算的函数。例如：

a = rand(1,1,3,1,2);

[b,n] = shiftdim(a);

b 是一个 3×1×2 的矩阵而 n 等于 2。

c = shiftdim(b,-n);

则 c=a

d = shiftdim(a,3);

d 是一个 1×2×1×1×3 的矩阵。

10. 删除单一行列

名称：squeeze

删除单一行列。

语法：B = squeeze(A)

描述：B = squeeze(A)返回一个和 A 有相同元素的数组 B，但所有的单一维都被去掉了。单一维是指任意满足 size(A,dim) = 1 的维。

举例：

考虑一个 2×1×3 的数组 A。这个数组有一个列的单一维，也就是说，它的每页只有一列。例如：

```
A(:,:,1) =          A(:,:,2) =
    1                   3
    2                   4
A(:,:,3) =
    5
    6
```

命令 B= squeeze(A)得到了一个 2×3 的矩阵。即：

```
B =
    1    3    5
    2    4    6
```

11. 下标的单一索引

名称：Sub2ind

下标的单一索引。

语法：该函数有如下两种表达形式：

- IND = sub2ind(siz,I,J)

- IND = sub2ind(siz,I1,I2,...,In)

描述：命令 sub2ind 确定了对应于一组下标值的等价的单一索引。

对于一个尺寸为 siz 的矩阵，IND = sub2ind(siz,I,J)返回了等价于数组 I 和 J 中行和列下标值的线形索引。

IND = sub2ind(siz,I1,I2,...,In)对于一个尺寸为 siz 的数组，返回了等价于数组 I1，I2，…，In 中 n 个下标的线形索引。

举例：

对于一个 2×2×2 的数组，由和下标等价的线形索引绘制的图如下：

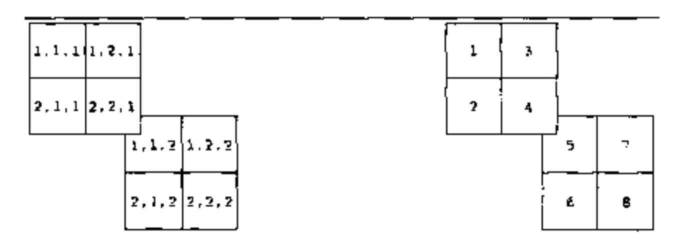


图 15-4　sub2 ind 的示例图

第十六章　图像可视化函数

16.1　基本绘图和图像函数

1. 垂直和水平直方图

名称：bar, barh

垂直和水平直方图。

语法：该函数有如下十种表达形式：

- bar(Y)
- bar(x,Y)
- bar(...,width)
- bar(...,'style')
- bar(...,LineSpec)
- [xb,yb] = bar(...)
- h = bar(...)
- barh(...)
- [xb,yb] = barh(...)
- h = barh(...)

描述：直方图将一个向量或矩阵的值描述为水平或垂直的直条。

bar(Y)为 Y 中的每一个元素画出一个直方。如果 Y 是一个矩阵，直方图由矩阵的每一行里的元素产生的直方所组成。当 Y 是向量时，X 轴的取值范围从 1 到 length(Y)，当 Y 是一个矩阵时，X 轴的取值范围是从 1 到 size(Y,1)，其中 size(Y,1)是行的数目。

bar(x,Y)在 x 中指定的位置处为 Y 中的每一个元素画出一个直方，这里 x 是 x 轴区间上的单调增加向量，它是为垂直直方图而定义的。如果 Y 是一个矩阵，bar 集成 Y 中的元素，这些元素位于与 x 中的一个元素相对应位置处的 Y 中的同样的行中。

bar(...,width)设置相对的直方宽度并且控制一个组里直条间的间距。默认宽度是 0.8，因此，如果用户没有指定 x，则组内的直条之间就会有一个小小的间距。如果宽度是 1，一个组内的棒棒就相互接触。

bar(...,'style')指定直方的样式。'style'有'group(组)'或'stack(堆栈)'两种取值。'group'是默认的显示模式。“group”显示 m 个垂直直方中的 n 个组，这里 n 是 Y 中行的数目，m 是 Y 中列的数目。对 Y 中每一列，组包含一个直条。“stack”为 Y 中的每一行显示一个直条。直条高度是行中元素的和。每一个直方图都是多颜色的，它对应于不同的元素并且显示每一个行元素对总和的相对贡献。

bar(...,LineSpec)通过使用由 LineSpec 确定的颜色来显示所有的直条。

[xb,yb] = bar(...)返回用户使用 plot(xb,yb)或 patch(xb,yb,C)所画出的向量。它将在关于图形的外观方面给用户提供更强的控制，例如，将一个直方图合并到更详细的绘图说明中。

h = bar(...)返回一个处理块图形对象的向量，在 Y 中，bar 为每一列创建一个块图形对象。

barh(...)，[xb,yb] = barh(...)和 h = barh(...)创建水平直方图。Y 确定直方的长度。向量 x 是 y 轴区间上的单调增加向量，它是为水平直方定义的。

举例：

(1) 画出一个钟形曲线。

```
x = -2:0.1:2;
bar(x,exp(-x.*x))
colormap hsv
```

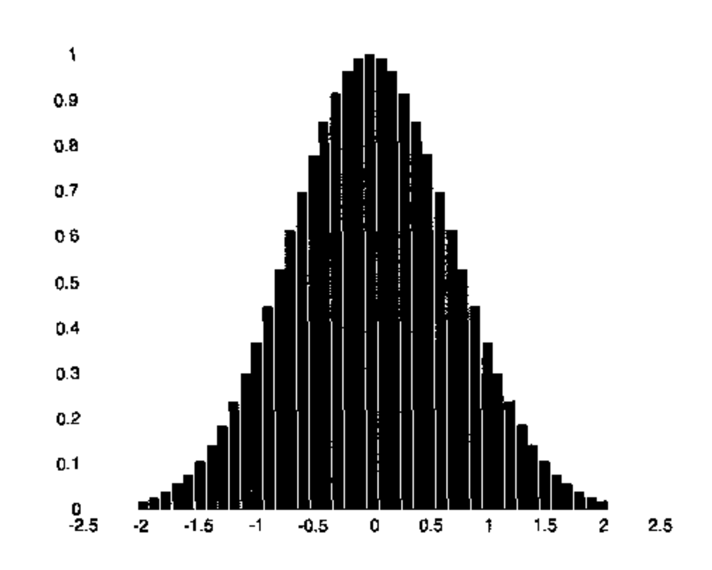


图 16-1 钟形曲线直方图

(2) 创建四个显示不同 bar 选项效应的子图。

```
Y = round(rand(5,3)*10);
subplot(2,2,1)
bar(Y,'group')
title 'Group'
subplot(2,2,2)
bar(Y,'stack')
title 'Stack'
subplot(2,2,3)
barh(Y,'stack')
title 'Stack'
subplot(2,2,4)
bar(Y,1.5)
```

title 'Width = 1.5'

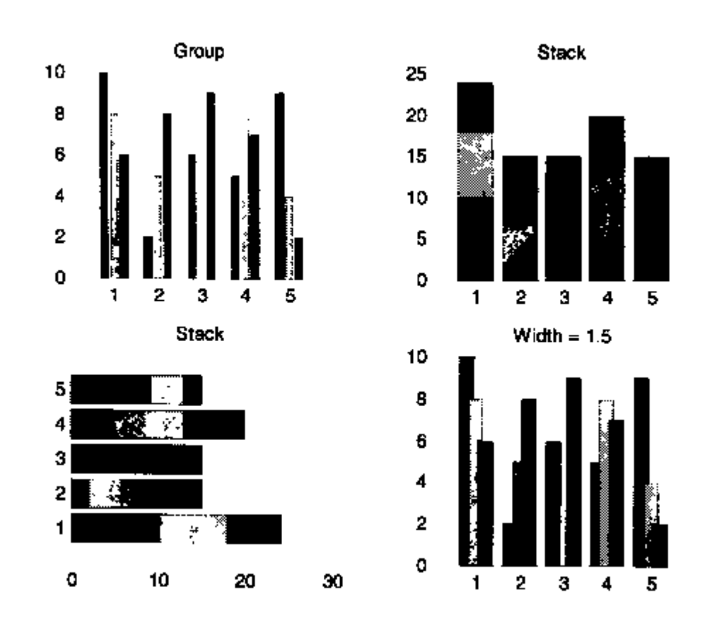


图 16-2　bar 选项效应的子图

2. 统计频数直方图

名称：hist

统计频数直方图。

语法：该函数有如下四种表达式：

- n = hist(Y)
- n = hist(Y,x)
- n = hist(Y,nbins)
- [n,xout] = hist(...)

描述：hist 绘制统计频数直方图。统计频数直方图显示了数据的分布状况。

n = hist(Y)用直方图表现在 10 个等分的区间上算得的 Y 的频数函数。如果 Y 是矩阵，则 hist 函数按照列的方向进行运算。

n = hist(Y,x)函数中的 x 是一个向量，用直方图在以 x 的各个分量为中心的区间上绘制计算得到的 Y 的频数函数。

n = hist(Y,nbins)用直方图表现在 nbins 个等分的区间上算得的 Y 的频数函数。

[n,xout] = hist(...)返回向量 n 和 xout，包含计算得到的 Y 的直方频数函数值和直方的位置。用户可以用 bar(xout,n)函数绘制直方图。

没有输出参数的 hist(...)函数直接绘制直方图。

举例：

(1) 从 Gaussian 数据中生成一个钟形曲线的直方图。

```
x = -3:0.1:3;
y = randn(1000,1);
hist(y,x)
```

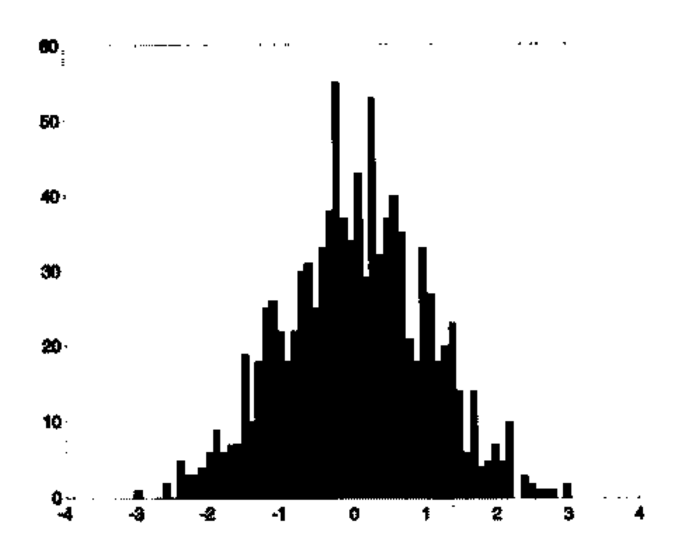


图 16-3　Gauss 分布钟形直方图

(2) 改变图形的颜色为红色。

```
h = findobj(gca,'Type','patch');
set(h,'FaceColor','r','EdgeColor','w')
```

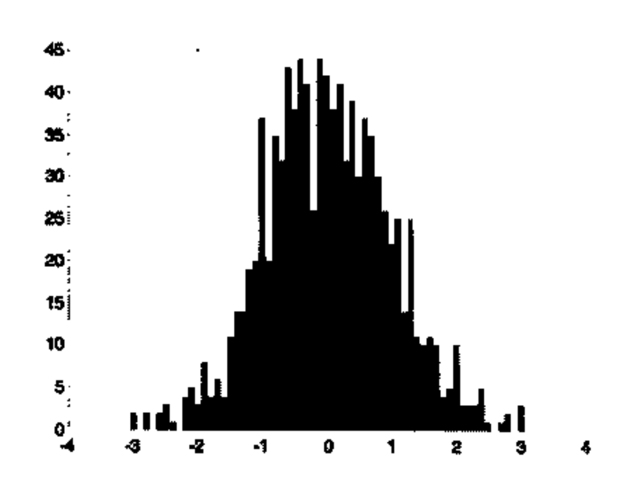


图 16-4　改变颜色后的钟形曲线直方图

3. *在图形窗口中保留当前图形*

名称：hold

在图形窗口中保留当前图形。

语法：该函数有如下三种表达形式：:

- hold on
- hold off
- hold

描述：hold 函数确定是否将一个新的图形对象加入到图形中或是否替换图形中的对象。hold on 保留当前的绘图和确定的轴的性质，以便使后续的图形命令加入到已经存在的

图形中。

hold off 在画出新图形之前将轴的性质重新设置为默认。hold off 是默认命令。

hold 在增加图形和替换图形之间切换 hold 状态。

4. 双对数刻度曲线图

名称：loglog

双对数刻度曲线图。

语法：该函数有如下五种表达形式：

- loglog(Y)
- loglog(X1,Y1,...)
- loglog(X1,Y1,LineSpec,...)
- loglog(...,'PropertyName',PropertyValue,...)
- h = loglog(...)

描述：如果 Y 包含实数，loglog(Y)绘出 Y 的列元素对下标的双对数曲线图。如果 Y 的列元素是复数，loglog(Y)便相当于 loglog(real(Y), imag(Y))，而对于其他情形，loglog 忽略虚部。

loglog(X1,Y1,...)绘制所有的 Xn 对 Yn 的坐标对。如果 Xn 或 Yn 是矩阵，loglog 绘制向量的分量对于矩阵的行或列的双对数刻度曲线图。

loglog(X1,Y1,LineSpec,...)绘制所有由 Xn、Yn、LineSpec 所定义的曲线，其中 LineSpec 决定了线型，标记符号和线的颜色。用户可以混合使用 Xn、Yn、LineSpec 三参数和 Xn、Yn 对：如：loglog(X1,Y1,X2,Y2,LineSpec,X3,Y3)。

loglog(...,'PropertyName',PropertyValue,...)设置由 loglog 函数所绘制的曲线对象的属性。

h = loglog(...)返回一个指向线对象的句柄列向量，每条线对应一个句柄。

举例：

绘制一个双对数刻度曲线图，标记符号为方块：

```
x = logspace(-1,1);
loglog(x,exp(x),'-s')
grid on
```

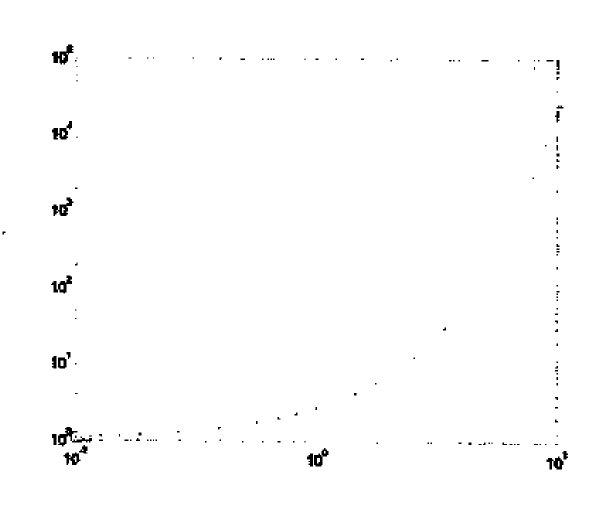


图 16-5　双对数刻度曲线

5. 饼图

名称：pie

饼图。

语法：该函数有如下三种表达形式：

- pie(X)
- pie(X,explode)
- h = pie(...)

描述：pie(X)使用 X 里的数据画出一张饼图。X 里的每一个元素被表示为饼图中的一张切片。pie(X,explode)分离饼图中的某一切片。explode 是一个对应于 X 的零与非零的向量或矩阵。如果是一个非零值，则从饼图的中心分离相应的切片，因此，如果 explode(i,j)是非零的，则 X(i,j)就是与中心分离的分支。explode 必须与 X 有相同的大小。

h = pie(...) 返回一个处理块和文本图形对象的向量

举例：

通过设置其相应的 explode 元素为 1 来强调第二个切片的分离。

```
x = [1 2 1 1 2 3];
explode=[0 1 0 0 0 0];
pie(x,explode)
```

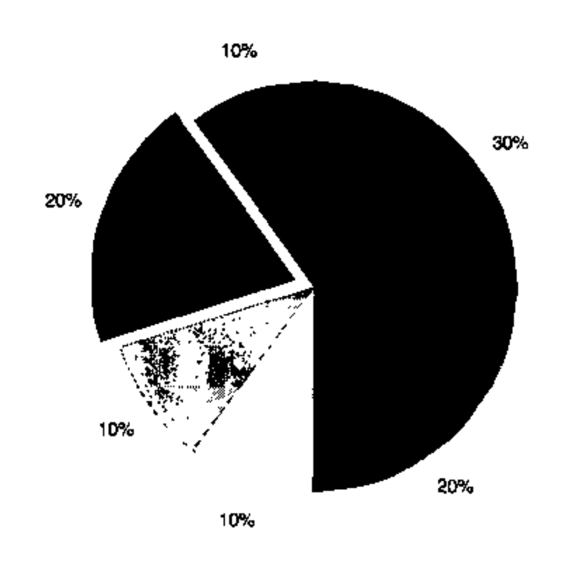


图 16-6　饼图

6. 绘制二维曲线图形

名称：plot 4

绘制二维曲线图形。

语法：该函数有如下五种表达形式：

- plot(Y)
- plot(X1,Y1,...)

- plot(X1,Y1,LineSpec,...)
- plot(...,'PropertyName',PropertyValue,...)
- h = plot(...)

描述：如果 Y 是实矩阵，则按列绘制每列元素值相对其下标的曲线图；如果 Y 是复数，plot(Y)等同于 plot(real(Y)和 imag(Y))。在其他情形，plot 函数忽略虚部。

plot(X1,Y1,...) 绘制所有关于 Xn 和 Yn 坐标对的曲线。如果 Xn 或 Yn 是矩阵，曲线以 X、Y 对应的行、列元素为横纵坐标分别绘制曲线，曲线的条数取决于矩阵的行或列。

plot(X1,Y1,LineSpec,...)绘制由 Xn，Yn，LineSpec 定义的所有曲线，其中 LineSpec 是控制曲线的线型、标记符号和色彩的参数组。

plot(...,'PropertyName',PropertyValue,...)设置由 plot 绘制的曲线图的各种属性。

h = plot(...)返回一个指向曲线对象句柄的列向量，每个曲线对应一个句柄。

举例：

(1) 下列语句可以得到如下的图形。

```
x = -pi:pi/10:pi;
y = tan(sin(x)) - sin(tan(x));
plot(x,y,'--rs','LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',10)
```

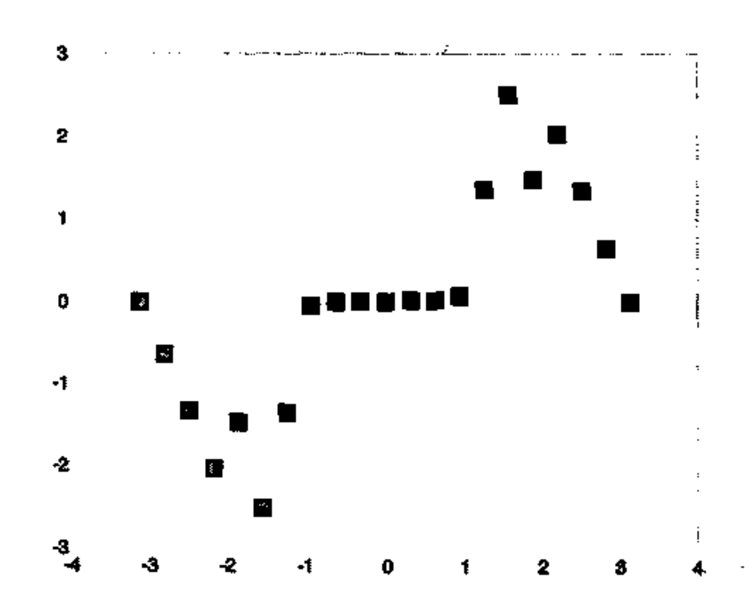


图 16-7 使用 plot 命令产生的图形

(2) 指定轴的刻度的位置和标记。例如，在下面绘制余弦函数曲线的图中对 x 轴进行了标注：

```
x = -pi:.1:pi;
y = sin(x);
```

```
plot(x,y)
set(gca,'XTick',-pi:pi/2:pi)
set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'})
xlabel('-\pi \leq \Theta \leq \pi')
ylabel('sin(\Theta)')
title('Plot of sin(\Theta)')
text(-pi/4,sin(-pi/4),'\leftarrow sin(-\pi\div4)',...
       'HorizontalAlignment','left')
```

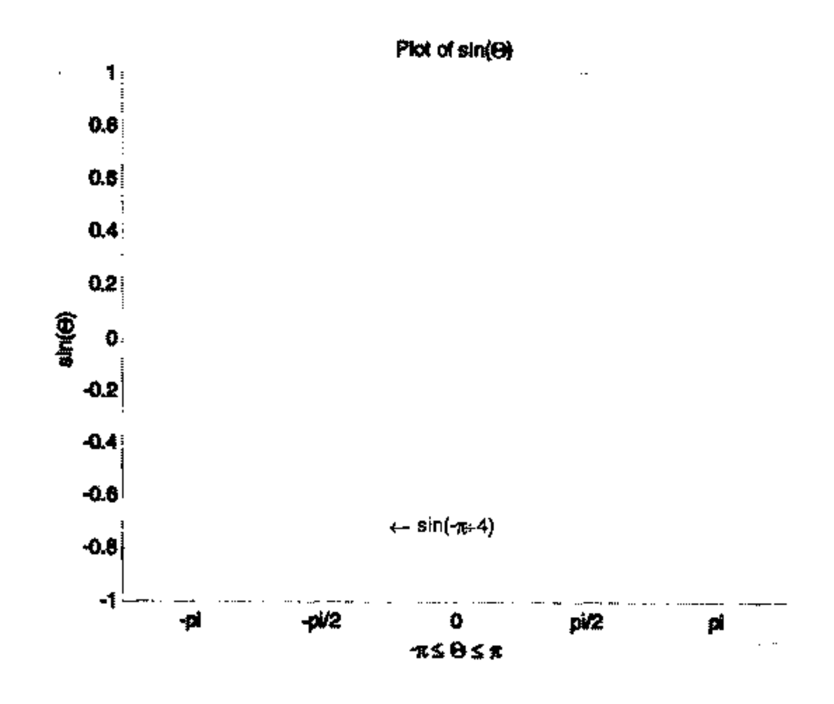


图 16-8　进行了标注的正弦曲线

7. 极坐标图

名称：polar

极坐标图。

语法：该函数有如下两种表达形式：

- polar(theta,rho)
- polar(theta,rho,LineSpec)

描述：polar 函数对于给出的极坐标，在笛卡尔平面里绘出极坐标图形和极坐标网格。

polar(theta,rho)创建一个幅角 theta 相对于半径 rho 的极坐标图，这里 theta 是从 x 轴到指定矢径的夹角，rho 是矢径的长度。

polar(theta,rho,LineSpec)中的 LineSpec 为在极坐标系下绘出的图形指定线型，绘图符号和颜色。

举例：

使用红色虚线绘出一个简单的极坐标图形。

```
t = 0:.02:2*pi;
```

polar(t,cos(4*t).*sin(4*t),'--r')

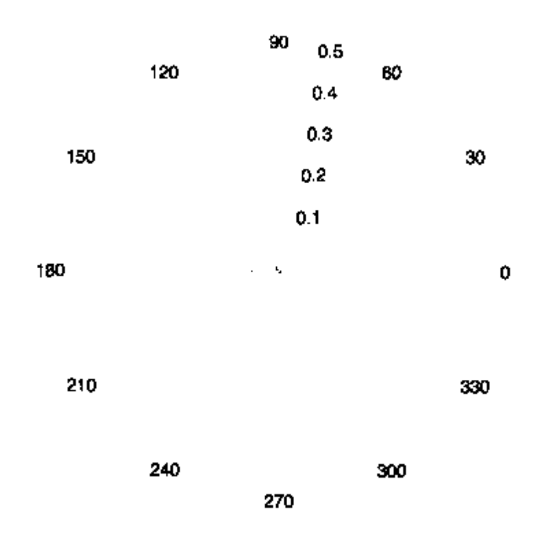


图 16-9 一个简单的极坐标图

8. 半对数刻度曲线图

名称：semilogx, semilogy

半对数刻度曲线图。

语法：该函数有如下七种表达形式：

- semilogx(Y)
- semilogx(X1,Y1,...)
- semilogx(X1,Y1,LineSpec,...)
- semilogx(...,'PropertyName',PropertyValue,...)
- h = semilogx(...)
- semilogy(...)
- h = semilogy(...)

描述：semilogx 和 semilogy 绘制相应于 x 轴和 y 轴的半对数刻度曲线。

semilogx(Y)生成以 10 为底对数刻度的 x 轴和线性刻度的 y 轴的半对数刻度曲线图。如果 Y 是实矩阵，则按列绘制每列元素值相对其下标的曲线图。如果 Y 是复数阵，semilogx(Y)等同于 semilogx(real(Y)，imag(Y))。对其他情形，semilogx 忽略虚部。

semilogx(X1,Y1,...)对 Xn 和 Yn 的坐标对，绘制所有的的曲线。如果 Xn 或 Yn 是矩阵，曲线以 X、Y 对应的行或列元素为横纵坐标分别绘制曲线，曲线的条数取决于矩阵的的行或列。

semilogx(X1,Y1,LineSpec,...)绘制由 Xn、Yn、LineSpec 定义的所有曲线，其中 LineSpec 是控制曲线的线型、标记和色彩的参数组。

semilogx(...,'PropertyName',PropertyValue,...)设置由 semilogx 绘制的曲线图的各种属性。

semilogy(…)生成以 10 为底对数刻度的 y 轴和线性刻度的 x 轴的半对数刻度曲线图。

h = semilogx(...)和 h = semilogy(...)返回一个指向曲线对象句柄的向量,每个曲线对应一个句柄。

举例:

绘出一个半对数刻度曲线图。

```
x = 0:.1:8;
semilogy(x,10.^x)
```

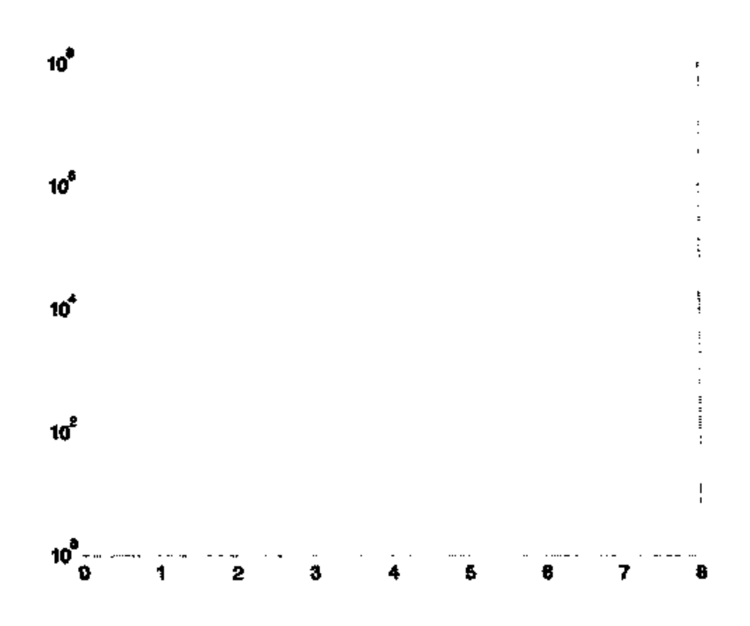


图 16-10　半对数刻度曲线图

9. 创建子图

名称：subplot

创建子图。

语法：该函数有如下四种表达形式：

- subplot(m,n,p)
- subplot(h)
- subplot('Position',[left bottom width height])
- h = subplot(...)

描述：subplot 将当前图形按行的方式划分为矩形块。每一个块包含一个坐标系，后续的图形是当前块的输出结果。

subplot(m,n,p)将一个图形划分为 m×n 个矩形块，在其中的第 p 个块里创建一个坐标系。新的坐标轴变为当前坐标轴。

subplot(h)使带有句柄 h 的轴成为当前轴，以便为后续的绘图命令使用。

subplot('Position',[left bottom width height])在由一个四元素向量确定的位置创建一个坐标轴. 左、底、宽和高位于从 0.0 到 0.1 的范围内的规范化的坐标系内。

h = subplot(...)将句柄返回至新的坐标轴。

举例:

绘出上半部分表示收入，下半部分表示支出的图形。

```
income = [3 4 5 6];
outgo = [2 3 35 4.9];
subplot(2,1,1); plot(income)
subplot(2,1,2); plot(outgo)
```

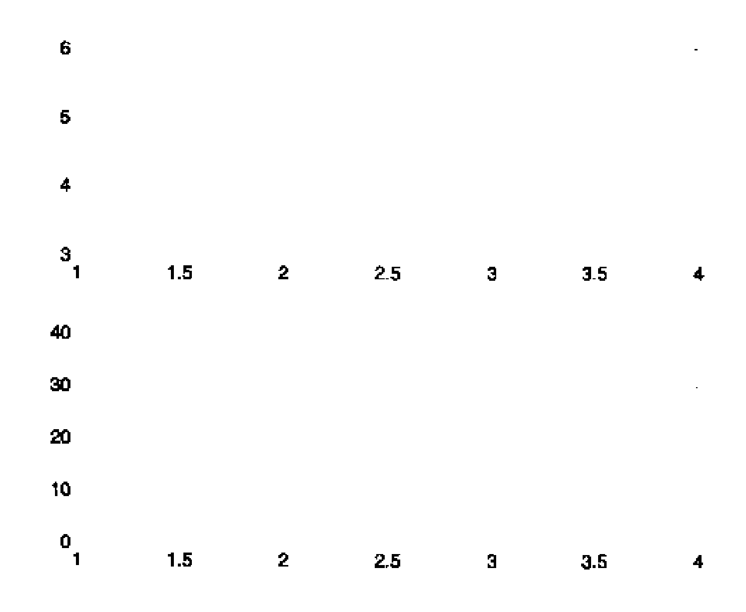


图 16-11　两个子图

16.2　三维绘图函数

1. 三维直方图

名称：bar3, bar3h

三维直方图。

语法：该函数有如下八种表达形式：

- bar3(Y)
- bar3(x,Y)
- bar3(...,width)
- bar3(...,'style')
- bar3(...,LineSpec)
- h = bar3(...)
- bar3h(...)
- h = bar3h(...)

描述：bar3 和 bar3h 绘制三维垂直和水平直方图。

bar3(Y)对给定的 Y 绘制三维的垂直直方图，Y 中的每个元素对应图中的一个直条。如

果 Y 是一个向量，则 x 方向的范围就是从 1 到 length(Y)。如果 Y 是一个矩阵，则 x 方向的范围就是从 1 到 size(Y,2)。

bar3(x,Y)在 x 指定的位置处绘制 Y 中所有元素的直方图。其中 x 是所有互不相等的分量的单调向量，它定义了每个直条在 Y 轴的间隔。如果 Y 是一个矩阵，bar3 在 x 的每行元素的数值所定义的位置上依次绘制所有的对应于 Y 矩阵相应行的所有列元素的直方图。

bar3(...,width)设置直条的宽度，从而控制同一组直条间的间隔。默认值是 0.8。如果用户没有定义 x 的值，则直条与直条间稍有空隙。如果定义宽度是 1，则直条与直条间便相互接触。

bar3(...,'style')设置直条的样式。样式有'detached'、'grouped'和'stacked'。默认值是'detached'。'detached'沿 x 方向依次排列 Y 的每一行的所有元素各自独立的直条。'grouped'显示 n 组直条，每组有 m 个直条，其中 n 是 Y 的行数，m 是 Y 的列数。每组中一列对应一个直条。'stacked'对 Y 的每一行绘制一个直条。直条的高度等于该行所有元素的和。每个直条有多种颜色组成，每种颜色对应于该行某个列元素及其在所有元素的和中所占的比例。

bar3(...,LineSpec)按 LineSpec 所定义的颜色绘制所有的直方图。

h = bar3(...)返回一个指向图形对象的句柄向量。bar3 对 Y 的每列生成一个对象。

bar3h(...)和 h = bar3h(...)绘制水平直方图。Y 确定直条的长度。单调向量 x 确定水平直条的 y 轴方向上的间隔。

举例：

绘制 6 幅图以显示 bar3 的各个参数的效果。Y 是由冷色图生成的 7×3 的矩阵。

```
Y = cool(7);
subplot(3,2,1)
bar3(Y,'detached')
title('Detached')
subplot(3,2,2)
bar3(Y,0.25,'detached')
title('Width = 0.25')
subplot(3,2,3)
bar3(Y,'grouped')
title('Grouped')
subplot(3,2,4)
bar3(Y,0.5,'grouped')
title('Width = 0.5')
subplot(3,2,5)
bar3(Y,'stacked')
title('Stacked')
subplot(3,2,6)
bar3(Y,0.3,'stacked')
title(`Width = 0.3')
colormap([1 0 0;0 1 0;0 0 1])
```

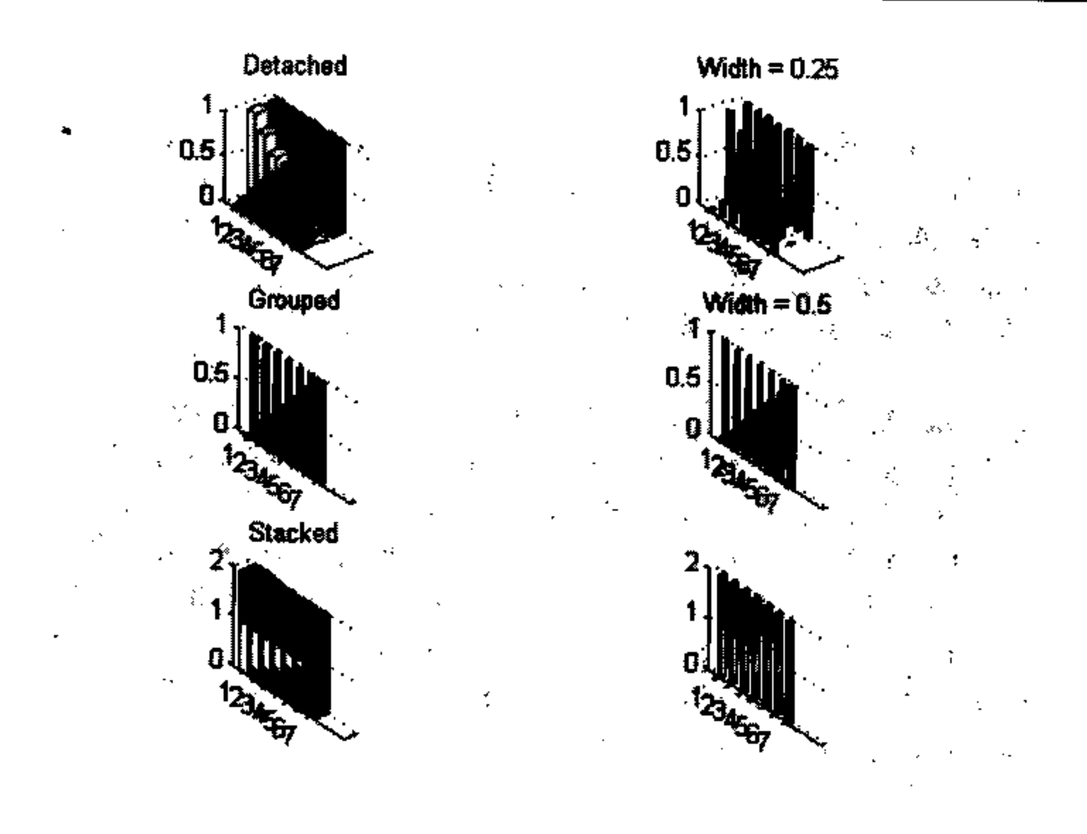


图 16-12 显示 bar3 的参数效果

2. 三维彗星图

名称：comet3

三维彗星图。

语法：该函数有如下三种表达形式：

- comet3(z)
- comet3(x,y,z)
- comet3(x,y,z,p)

描述：一个 comet 绘图命令是一个动画，其中有一个圆(彗星头)会在屏幕上追踪数据点。彗星体是跟随彗星头的轨迹段。彗星尾是一条追踪整个函数的实心线。

comet3(z)是显示向量 z 的一个三维彗星状绘图。

comet3(x,y,z)是显示通过点[x(i),y(i),z(i)]的曲线的一个彗星状绘图。

comet3(x,y,z,p)确定一个长度为 p*length(y)的彗星体。

举例：

创建一个三维彗星图。

```
t =-10*pi:pi/250:10*pi;
comet3((cos(2*t).^2).*sin(t),(sin(2*t).^2).*cos(t),t)
```

3. 绘制柱面图

名称：cylinder

绘制柱面图。

语法：该函数有如下四种表达形式：

- [X,Y,Z] = cylinder
- [X,Y,Z] = cylinder(r)
- [X,Y,Z] = cylinder(r,n)
- cylinder(...)

描述：cylinder 生成一个单位柱面上点的 x、y、z 的坐标值。所得的 x、y、z 的坐标值可以通过 surf 或 mesh 命令，或直接利用无输出参数的 cylinder 命令表现为图形。

[X,Y,Z] = cylinder 返回矢径为 1 的柱面的 x、y、z 的坐标值。该柱面的圆周上有 20 个等间距的分割线条。[X,Y,Z] = cylinder(r)返回由矢径 r 定义的母线所对应的柱面的 x，y，z 的坐标值。“母线”向量 r 总被看作是在单位高度里等分刻度上定义的半径向量。该柱面的圆周上有 20 个等间距的分割线条。[X,Y,Z] = cylinder(r,n) 返回由矢径 r 定义的母线所对应的柱面的 x、y、z 的坐标值。该柱面的圆周上有 n 个等间距的分割线条。

cylinder(...)无参数输出，等效于用 surf 绘制柱面图。

举例：

(1) 生成一个表面颜色随机分布的柱面图。

```
cylinder
axis square
h = findobj('Type','surface');
set(h,'CData',rand(size(get(h,'CData'))))
```

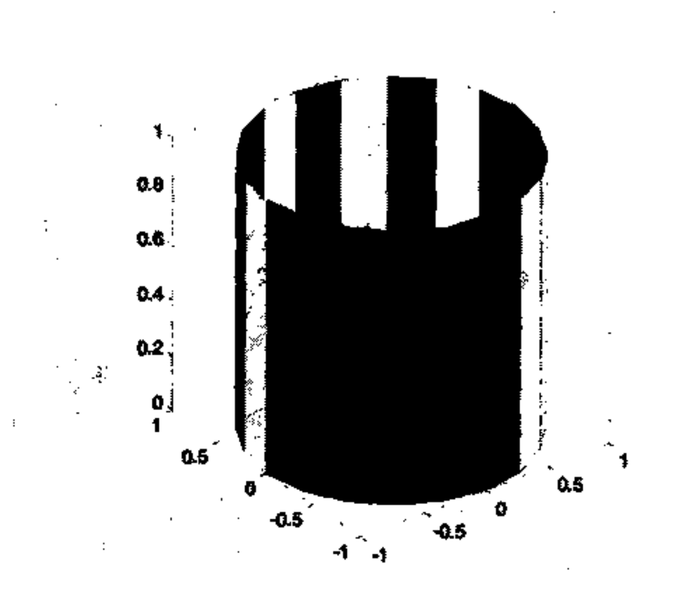


图 16-13 母线是直线的柱面图

(2) 生成一个母线由函数 2+cos(t)定义的柱面图。

```
t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(2+cos(t));
surf(X,Y,Z)
axis square
```

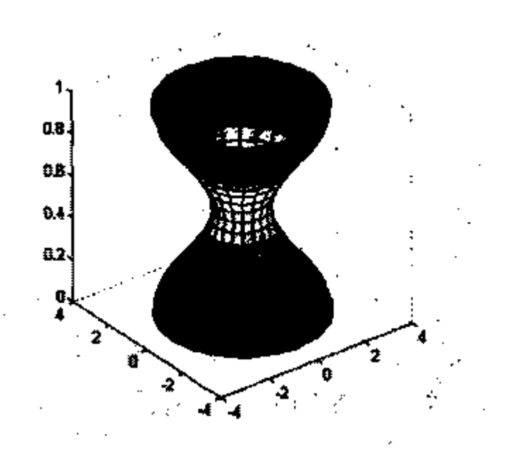


图 16-14　母线是曲线的柱面图

4. 填充的三维多边形

名称：fill3

填充的三维多边形。

语法：该函数有如下五种表达形式：

- fill3(X,Y,Z,C)
- fill3(X,Y,Z,ColorSpec)
- fill3(X1,Y1,Z1,C1,X2,Y2,Z2,C2,...)
- fill3(...,'PropertyName',PropertyValue)
- h = fill3(...)

描述：fill3 函数用浅色阴影和深色阴影填充空间多边形。

fill3(X,Y,Z,C)填充三维多边形。X、Y 和 Z 确定了多边形的顶点。如果 X、Y 或 Z 是一个矩阵，fill3 创建一个 n 边形，其中 n 是矩阵中列的数目。必要时，fill3 通过将最后一个顶点连接到第一个顶点来使多边形闭合。

C 指定颜色，其中 C 是当前色图里的一个索引向量或矩阵。如果 C 是一个行向量，length(C)必须等于 size(X,2)和 size(Y,2)；如果 C 是一个列向量，length(C)必须等于 size(X,1)和 size(Y,1)。

fill3(X,Y,Z,ColorSpec)填充一个三维多边形，这个多边形是用带有 ColorSpec 来确定颜色的 X、Y 和 Z 所定义。

fill3(X1,Y1,Z1,C1,X2,Y2,Z2,C2,...)确定多重填充的三维区域。

fill3(...,'PropertyName',PropertyValue)允许用户为特殊的块性质设置值。

h = fill3(...)对块图对象返回一个句柄向量，每一个块对应一个句柄。

举例：

用插值颜色创建四个填充的三角形

```
X = [0 1 1 2;1 1 2 2;1 1 1 1];
Y = [1 1 1 1;1 0 1 0;0 0 0 0];
Z = [1 1 1 1;1 0 1 0;0 0 0 0];
C = [0.5000 1.0000 1.0000 0.5000;
     1.0000 0.5000 0.5000 0.1667;
```

```
    0.3330 0.3330 0.5000 0.5000];
fill3(X,Y,Z,C)
```

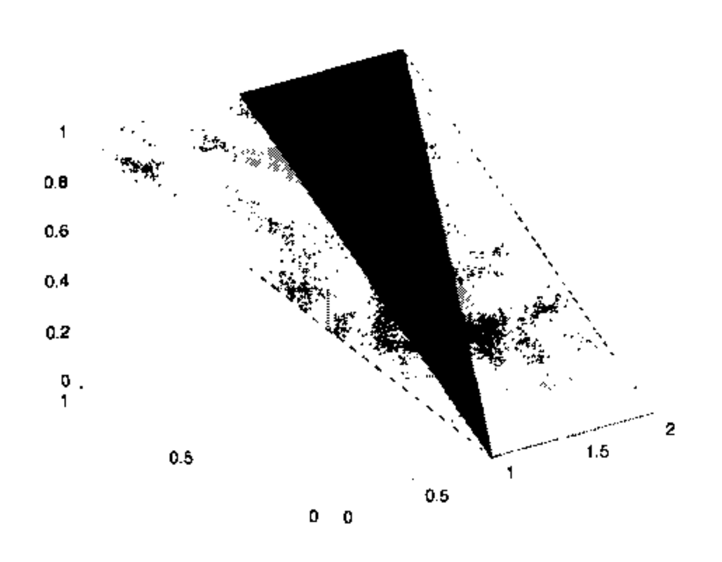


图 16-15　填充三角形

5. 三维直角坐标曲线图

名称：plot3

三维直角坐标曲线图。

语法：该函数有如下四种表达形式：

- plot3(X1,Y1,Z1,...)
- plot3(X1,Y1,Z1,LineSpec,...)
- plot3(...,'PropertyName',PropertyValue,...)
- h = plot3(...)

描述：plot3 函数绘制一组数据点的三维曲线图。

plot3(X1,Y1,Z1,...)对于给出的向量或者矩阵 X1、Y1、Z1 在三维空间中绘制一条或多条曲线，曲线上点的坐标由 X1、Y1、Z1 中的元素确定。

plot3(X1,Y1,Z1,LineSpec,...)绘制由 X1、Y1、Z1 和 LineSpec 定义的曲线，其中 LineSpec 确定了曲线的属性：线型，标记符号和曲线的颜色。

plot3(...,'PropertyName',PropertyValue,...)设置所有绘制的曲线图的各种属性。

h = plot3(...)返回一个指向曲线图中对象的列向量句柄，每条曲线对应一个句柄。

举例：

绘制一个三维螺旋线。

```
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t)
grid on
axis square
```

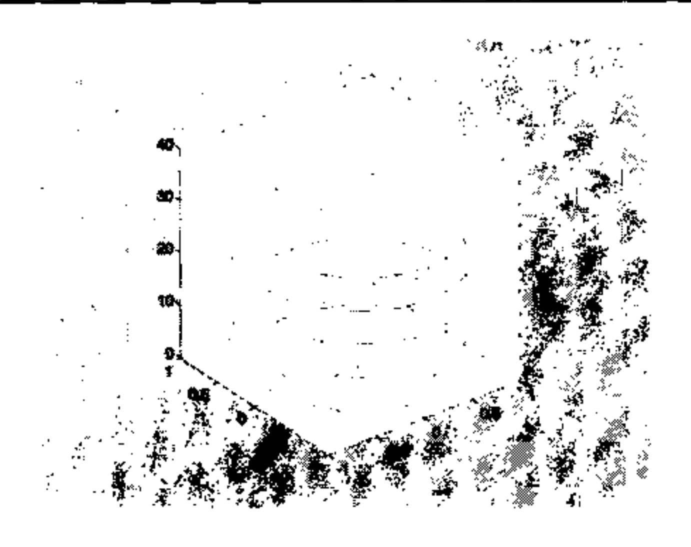


图 16-16　三维螺旋线

6. 三维向量场图

名称：quiver3

三维向量场图。

语法：该函数有如下六种表达形式：

- quiver3(Z,U,V,W)
- quiver3(X,Y,Z,U,V,W)
- quiver3(...,scale)
- quiver3(...,LineSpec)
- quiver3(...,LineSpec,'filled')
- h = quiver3(...)

描述：一个三维向量场图显示在点(x,y,z)处的分量为(u,v,w)的向量。

quiver3(Z,U,V,W)在由矩阵 Z 确定的等空间表面点处绘出向量。为防止重叠，quiver3 根据向量之间的距离自动调整并绘出向量。

quiver3(X,Y,Z,U,V,W)在点(x,y,z)处绘出分量为(u,v,w)的向量。矩阵 X、Y、Z、U、V、W 必须具有相同的阶数，并且包含对应的位置和向量分量。

quiver3(...,scale)自动调整并绘出向量，以防止重叠，然后乘以所给的比例。scale = 2 使其相对的长度加倍，scale = 0.5 使其相对的长度减半。如果不进行自动的比例调整，则使用 scale = 0 绘出向量。

quiver3(...,LineSpec)使用任意有效的 LineSpec 确定线型和颜色。

quiver3(...,LineSpec,'filled')填充由 LineSpec 确定的标记。

h = quiver3(...)返回一个线句柄向量。

举例：

画出函数 z = x*exp(-x*x-y*y)的表面法向。

```
[X,Y] = meshgrid(-2:0.25:2,-1:0.2:1);
Z = X.* exp(-X.^2 - Y.^2);
[U,V,W] = surfnorm(X,Y,Z);
quiver3(X,Y,Z,U,V,W,0.5);
```

```
hold on
surf(X,Y,Z);
colormap hsv
view(-35,45)
axis ([-2 2 -1 1 -.6 .6])
hold off
```

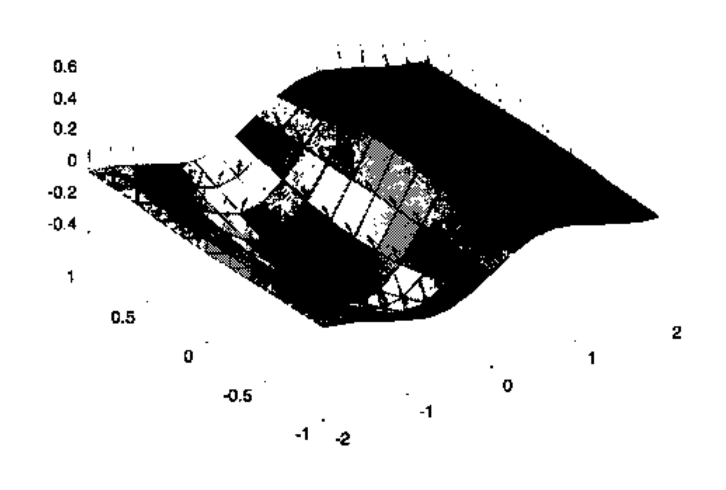


图 16-17　一个函数的法向表面

7. 切片图

名称：slice

切片图。

语法：该函数有如下六种表达形式：

- slice(V,sx,sy,sz)
- slice(X,Y,Z,V,sx,sy,sz)
- slice(V,XI,YI,ZI)
- slice(X,Y,Z,V,XI,YI,ZI)
- slice(...,'method')
- h = slice(...)

描述：slice 绘制立体空间的正交切片图。

slice(V,sx,sy,sz)在向量 sx、sy 和 sz.所定义的点上沿 x、y、z 方向绘制三维立体空间 V 的切片。V 是 m×n×p 的三元函数值数组，默认值是 X=1：n、Y=1：n，Z=1：p。向量 sx，sy 和 sz 中的每个元素都在 x，y 或 z 方向上定义了一个切平面。

slice(X,Y,Z,V,sx,sy,sz)绘制由 X、Y 和 Z 确定的三维立体空间数组 V 的切面。X、Y 和 Z 必须是单调、正交的（如 meshgrid 函数所生成的）。每点的颜色由对 V 的三维插值方法决定。

slice(V,XI,YI,ZI)在体 V 中绘制由 XI、YI 和 ZI 确定的切面，其中矩阵 XI、YI 和 ZI 定义了一个表面。体在表面点处被计算。XI、YI 和 ZI 必须有相同阶数。

slice(X,Y,Z,V,XI,YI,ZI)绘制 V 的多个由数组 XI，YI，ZI 确定的切面。

slice(...,'method')指定了插值方法。'method'包括'linear', 'cubic'，和'nearest'三种方法。linear 指定三点线性插值法（默认值），cubic 指定立方插值法，nearest 指定最邻近点插值。

h = slice(...) 返回一个指向曲面对象的向量句柄。

举例：

(1) 在$-1 \le x \le 1$，$-1 \le y \le 1$，$-1 \le z \le 1$ 的范围内可视化给定的函数 v = x*exp*(-x*x-y*y-z*z)。

```
[x,y,z] = meshgrid(-1:.1:1,-1:.1:1,-1:.1:1);
v = x.*exp(-x.^2-y.^2-z.^2);
xslice = [-1.,.8,1]; yslice = 1; zslice = [-1,0];
slice(x,y,z,v,xslice,yslice,zslice)
colormap hsv
```

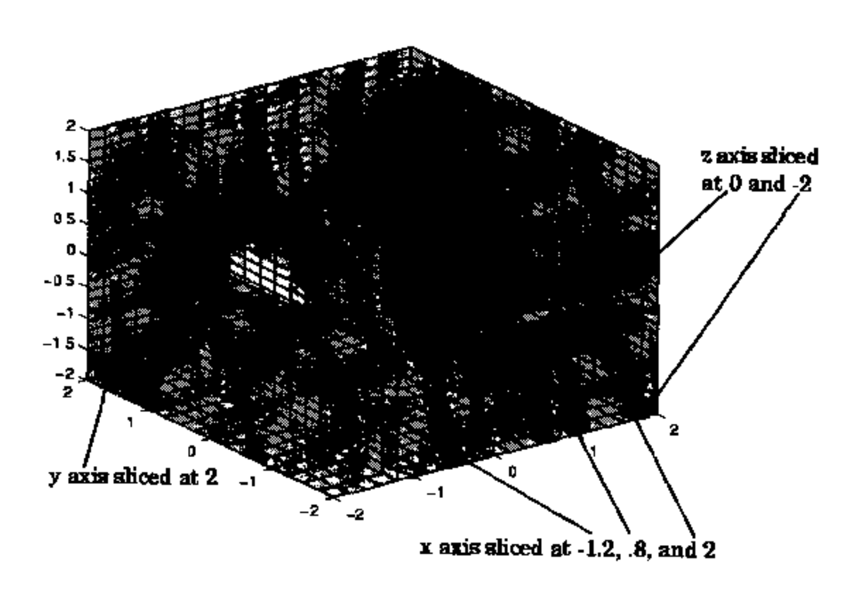


图 16-18　函数可视化

(2) 以任意角度作切面。下列语句可实现用一个旋转的切平面去切上例中的立体空间。利用 for 循环实现切面沿 z 轴一边移动一边与实体空间相切。

```
for i = -2:.5:2
    hsp = surf(linspace(-2,2,20),linspace(-2,2,20),zeros(20)+i);
    rotate(hsp,[1,-1,1],30)
    xd = get(hsp,'XData');
    yd = get(hsp,'YData');
    zd = get(hsp,'ZData');
    delete(hsp)
    slice(x,y,z,v,[-2,2],2,-2) % Draw some volume boundaries
    hold on
    slice(x,y,z,v,xd,yd,zd)
    hold off
```

```
    axis tight
    view(-5,10)
    drawnow
end
```

下图示例了同一切面在通过立体空间的过程中的三个位置时的切面图。

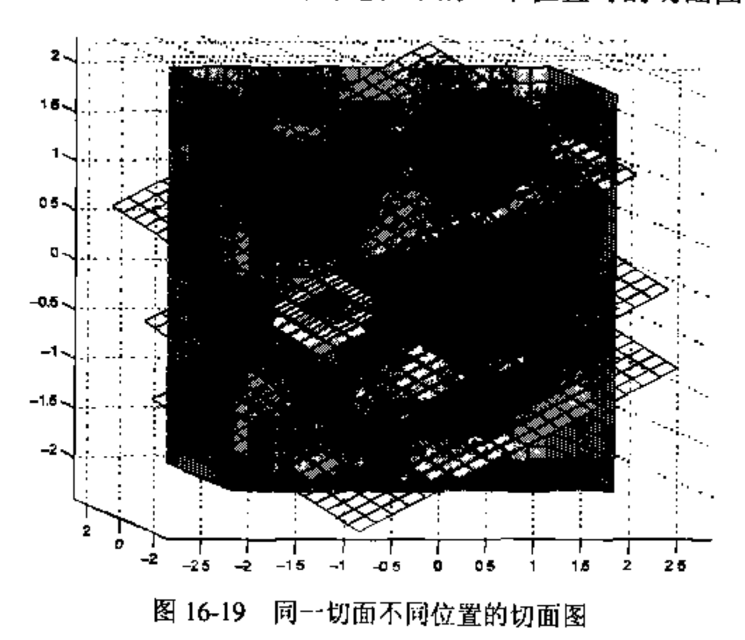


图 16-19　同一切面不同位置的切面图

(3)　非平面切面的切面图。下例实现了一个球面穿越上面例子中的立体空间时的切面图。

```
[xsp,ysp,zsp] = sphere;
slice(x,y,z,v,[-2,2],2,-2)   % Draw some volume boundaries
for i = -3:.2:3
    hsp = surface(xsp+i,ysp,zsp);
    rotate(hsp,[1 0 0],90)
    xd = get(hsp,'XData');
    yd = get(hsp,'YData');
    zd = get(hsp,'ZData');
    delete(hsp)
    hold on
    hslicer = slice(x,y,z,v,xd,yd,zd);
    axis tight
    xlim([-3,3])
    view(-10,35)
    drawnow
    delete(hslicer)
    hold off
end
```

下图显示了球面在穿越过程中的三个位置时的切面图。

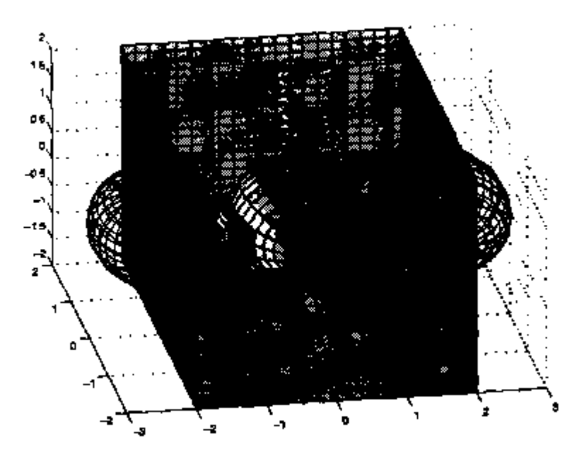


图 16-20 球面的三个不同位置的切面图

8. 生成球面

名称：sphere

生成球面。

语法：该函数有如下三种表达形式：

- sphere
- sphere(n)
- [X,Y,Z] = sphere(...)

描述：sphere 函数为使用 surf 和 mesh 命令而生成 x、y 和 z 坐标下的单位球面。sphere 生成由 20×20 个面组成的一个球面。sphere(n)在当前图形中画出有一个 n×n 个面的球面图。

[X,Y,Z] = sphere(n)在带有(n+1)×(n+1)刻度的三维笛卡尔系内返回一个球面的坐标值。用户可以用 surf(X,Y,Z)或 mesh(X,Y,Z)命令画出球面。

举例：

生成并绘制一个球面。

```
sphere
axis equal
```

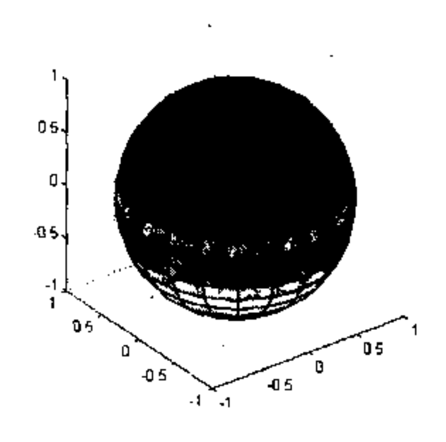


图 16-21 一个具有 20×20 个面的球面

9. 三维火柴杆图

名称：stem3

三维火柴杆图。

语法：该函数有如下五种表达形式：

- stem3(Z)
- stem3(X,Y,Z)
- stem3(...,'fill')
- stem3(...,LineSpec)
- h = stem3(...)

描述：三维 stem 函数绘制从 xy 平面上伸展出的火柴杆直线。每个火柴杆直线由一个圆圈（默认值）或其他的标记符号终止，符号在 z 轴方向上的位置代表了其数值的大小。

stem3(Z)依据数据序列 Z 的大小绘制从 xy 平面上伸展的火柴杆直线。x 和 y 是自动生成的。当 Z 是行向量时，stem3 在相同的 y 值上等间距分布的 x 值处绘制所有元素的火柴杆直线；当 Z 是列向量时，stem3 在相同的 x 值上等间距分布的 y 值处绘制所有元素的火柴杆直线。

stem3(X,Y,Z)在由 X 和 Y 指定的位置处绘制数据序列 Z，而且 X、Y 和 Z 必须是同维数的向量或矩阵。

stem3(...,'fill')指定了是否在火柴杆直线末端的圆圈内填充颜色。

stem3(...,LineSpec)为绘制的火柴杆直线指定线型，标记符号和颜色。

h = stem3(...)返回指向火柴杆直线图形对象的句柄。

举例：

通过生成三维火柴杆图可视化一个二变量函数。

```
X = linspace(0,1,10);
Y = X./4;
Z = 2*sin(X) + cos(Y);
stem3(X,Y,Z,'fill')
view(-30,30)
```

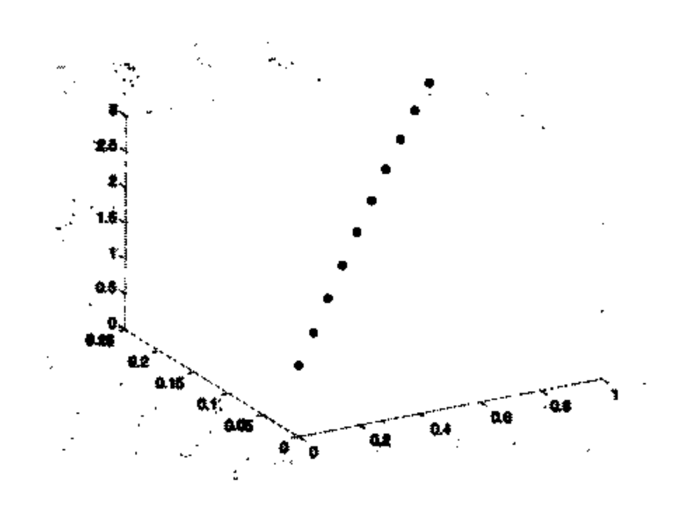


图 16-22　三维火柴杆图

10. 瀑布水线图

名称：waterfall

瀑布水线图。

语法：该函数有如下四种表达形式：

- waterfall(Z)
- waterfall(X,Y,Z)
- waterfall(...,C)
- h = waterfall(...)

描述：waterfall 函数画出与 meshz 函数相似的网线，但是它不能从矩阵的列中生成线，这就产生了一个瀑布效应。

waterfall(Z)使用 x = 1:size(Z,1)和 y = 1:size(Z,1)创建一个瀑布水线图。Z 确定颜色，颜色是与表面高度成比例的。

waterfall(X,Y,Z)使用在 X、Y 和 Z 里确定的值创建一个瀑布水线图，Z 仍然确定颜色，颜色是与表面高度成比例的。如果 X 和 Y 是向量，X 对应 Z 的列，Y 对应行，这里 length(x) = n、length(y) = m，[m,n] = size(Z)。X 和 Y 是向量或矩阵，它定义了图形的 x 坐标和 y 坐标。Z 是定义了图形 z 坐标的矩阵(如高于平面的高度)。如果 C 被忽略，颜色与 Z 成比例。

waterfall(...,C)用有等级的颜色值从当前色图中获得颜色。颜色的等级是由 C 的范围确定的，它必须与 Z 有相同的值。为从当前色图中获得颜色，MATLAB 执行一个 C 上的线性变换。

h = waterfall(...)返回一个用于绘图的块图对象句柄。

举例：

产生 peaks 函数的一个瀑布水线图。

[X,Y,Z] = peaks(50);

waterfall(X,Y,Z)

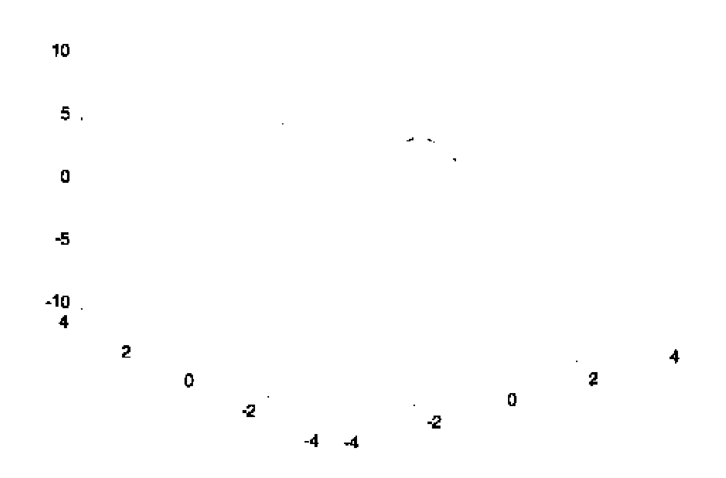


图 16-23 peaks 函数的瀑布水线图

16.3 绘制标注和网格

1. 为等高线图的等高线加高度数值标记

名称：clabel

为等高线图的等高线加高度数值标记。

语法：该函数有如下六种表达形式：

- clabel(C,h)
- clabel(C,h,v)
- clabel(C,h,'manual')
- clabel(C)
- clabel(C,v)
- clabel(C,'manual')

描述：clabel 函数为二维等高线图加上高度数值标记。

clabel(C,h)把标记旋转适当角度并插入到等高线上。该函数根据等高线图的大小只在该图上插入适当的标记。

clabel(C,h,v)只在向量 v 给定的高度值对应的等高线插入标记。

clabel(C,h,'manual')只在用户用鼠标或键盘所指定的位置处插入标记。按下鼠标左键或空格键，则在鼠标或光标最近的等高线上插入标记。当光标在图形窗口中时，按回车键后结束插入标记。所有插入的标记都经过适当的旋转插入到等高线上。

clabel(C)根据等高线函数 contour 输出的等高线结构 C 的定义，在当前的等高线图上插入标记。该函数标记所有的等高线图，并且标记的位置是随机产生的。

clabel(C,v)只在向量 v 指定的位置上插入 C 所定义的标记。

clabel(C,'manual')只在用户选择的位置上插入 C 所定义的标记。

举例：

生成，绘制并标记一个简单的等高线图。

```
[x,y] = meshgrid(-2:.2:2);
z = x.^exp(-x.^2-y.^2);
[C,h] = contour(x,y,z);
clabel(C,h)
```

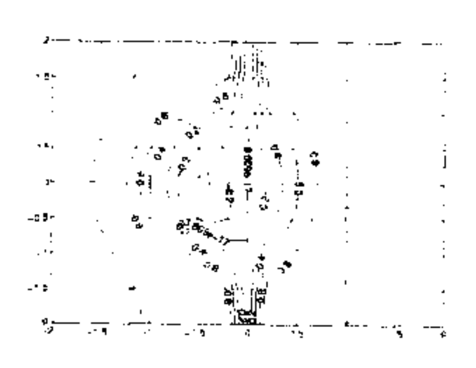


图 16-24 一个带标记的等高线图

2. 使用日期标注标记线

名称：datetick

使用日期标注标记线。

语法：该函数有如下两种表达形式：

- datetick(tickaxis)
- datetick(tickaxis,dateform)

描述：datetick(tickaxis)使用数据标记一个坐标轴的标记线，替换默认的数值标签。tickaxis是'x'、'y'或'z'字符串。默认是'x'。datetick 根据指定轴的最小限和最大限来选择一个标签格式。

datetick(tickaxis,dateform)根据整型 dateform(见 table)来格式化标签。为了产生当前结果，用于指定轴的数据必须是连续的日期数(如由 datenum 所产生的一样)。

表 16-1　　使用日期标注标记线表

日期形式(Dateform)	格式(Format)	例子(Example)
0	day-month-year hour:minute	01-Mar-1995 03:45
1	day-month-year	01-Mar-1995
2	month/day/year	03/01/95
3	month, 三个字母	Mar
4	month, 一个字母	M
5	month, 数字	3
6	month/day	03/01
7	day of month	1
8	day of week, 三个字母	Wed
9	day of week, 单个字母	W
10	year, 四位数字	1995
11	year, 两位数字	95
12	month year	Mar95
13	hour:minute:second	15:45:17
14	hour:minute:second AM or PM	03:45:17
15	hour:minute	15:45
16	hour:minute AM or PM	03:45 PM

举例：

考虑图示的基于 1990 年美国人口普查的人口数据。

```
% Time interval
t = (1900:10:1990)';
% Population
p = [75.995 91.972 105.711 123.203 131.669 ...
     150.697 179.323 203.212 226.505 249.633]';
% Convert years to date numbers and plot
plot(datenum(t,1,1),p)
grid on
% Replace x-axis ticks with 2-digit year labels
datetick('x',11)
```

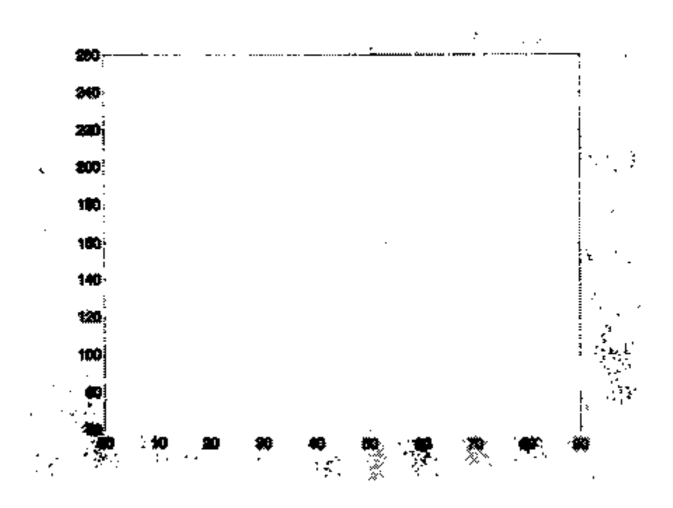


图 16-25　人口普查数据图

3. 控制二维和三维图形的网格

名称：grid

控制二维和三维图形的网格。

语法：该函数有如下三种表达形式：

- grid on
- grid off
- grid

描述：grid 函数控制当前坐标轴的网格线的显示和关闭。

grid on 在当前坐标轴上添加网格线。

grid off 从当前坐标轴隐藏网格线。

grid 切换网格线的可见是否的状态。

4. 使用鼠标确定文本在二维视图中的位置

名称：gtext

使用鼠标确定文本在二维视图中的位置。

语法：该函数有如下两种表达形式：

- gtext('string')
- h = gtext('string')

描述：当用户使用鼠标选择一个位置后，gtext 在该位置处显示当前图形窗口的一个文本串。

当鼠标指针位于图形窗口内时，gtext('string')等待用户按住鼠标按钮或键盘键。按鼠标按钮或键盘的任何键将把'string'放置在图形中的所选位置处。

在用户将'string'放置在图形中的所选位置处后，h = gtext('string')返回一个指向文本图形对象的句柄。

举例：

将一个标签放置在当前图形上。

gtext('Note this!')

5. 在图形上显示图例

名称：legend

在图形上显示图例。

语法：该函数有如下十种表达形式：

- legend('string1','string2',...)
- legend(h,'string1','string2',...)
- legend(string_matrix)
- legend(h,string_matrix)
- legend(axes_handle,...)
- legend('off')
- legend(h,...)
- legend(...,pos)
- h = legend(...)
- [legend_handle,object_handles] = legend(...)

描述：legend 在各种类型的图形（曲线图，直条图，饼图等）上设置图例。对于图形上的每一条曲线，legend 显示一个该曲线线型，标记符号及颜色的示例，并且标注上用户指定的说明文字。对于可填充区域（块或物体表面），legend 显示该表面元素的样板及说明文字。

legend('string1','string2',...)在当前坐标轴上显示用户指定的文本串，以标记各组数据。

legend(h,'string1','string2',...)对句柄向量 h 中定义的图形根据用户定义的文本串设置该图形的标记。

legend(string_matrix) 用矩阵 string_matrix 中各行的文本串作为图例。等效于 legend(string_matrix(1,:),string_matrix(2,:),...)。

legend(h,string_matrix)把矩阵 string_matrix 中各行的文本串作为句柄向量 h 中定义的相应图形的图例。

legend(axes_handle,...)显示由 axes_hadle 句柄所指定的坐标轴的图例。

legend('off'),legend(axes_handle,'off')去除当前坐标轴或 axes_handle 所指定的坐标轴的图例。

legend_handle = legend 返回指向当前坐标轴的图例的句柄。如果没有图例，则返回一个空向量。

无参数的 legend 刷新当前图形上的所有图例。

legend(legend_handle)只刷新指定的图例。

legend(...,pos)按 pos 的定义放置图例。pos = -1 在轴的外边界靠右放置图例；pos = 0 在轴的内边界且最大限度的不遮挡数据点的位置放置图例；pos = 1 在轴的右上角放置图例（默认值）；pos = 2 在轴的左上角放置图例；pos = 3 在轴的左下角放置图例；pos = 4 在轴的右下角放置图例。

[legend_handle,object_handles] = legend(...)返回(legend_handle)的句柄，该句柄指向的对象是一种轴对象，包括图例中所用到的线，块和文本等对象的句柄。这些句柄有助于用户修改相应对象的属性。

举例：

在一幅 sine 和 cosine 图上添加图例：

```
x = -pi:pi/20:pi;
plot(x,cos(x),'-ro',x,sin(x),'-.b')
h = legend('cos','sin',2);
```

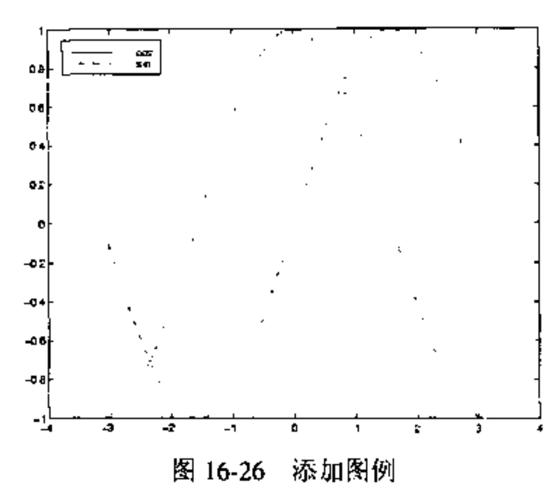


图 16-26　添加图例

在此例中，plot 指令定义 cosine 函数的线型：实线，红色；sine 函数的线型：点划线，蓝色。

6. *使用左右双 y 轴创建图形*

名称：plotyy

使用左右双 y 轴创建图形。

语法：该函数有如下四种表达形式

- plotyy(X1,Y1,X2,Y2)
- plotyy(X1,Y1,X2,Y2,'function')
- plotyy(X1,Y1,X2,Y2,'function1','function2')
- [AX,H1,H2] = plotyy(...)

描述：plotyy(X1,Y1,X2,Y2)用左边的 y 轴画出 X1 对应于 Y1 的图，用右边的 y 轴画出 X2 对应于 Y2 的图。

plotyy(X1,Y1,X2,Y2,'function')是使用由字符串'function'指定的绘图函数,而不是通过绘图产生每一个图形。'function'可以是 plot、semilogx、semilogy、stem 或任何满足 h = function(x,y) 的 MATLAB 函数。

plotyy(X1,Y1,X2,Y2,'function1','function2')使用 function1(X1,Y1)为左轴画出图形，使用 function1(X2,Y2)为右轴画出图形。

[AX,H1,H2] = plotyy(...)返回在 AX 里创建的两个轴的句柄和来自 H1 和 H2 里的每一个图形中的图对象句柄。AX(1)是左轴，AX(2)是右轴。

举例：

使用 plot 函数和两个 y 轴创建一个图形。

```
t = 0:pi/20:2*pi;
y1 = cos(t);
y2 = 0.5*sin(t-1.5);
```

plotyy(t,y1,t,y2,'plot')

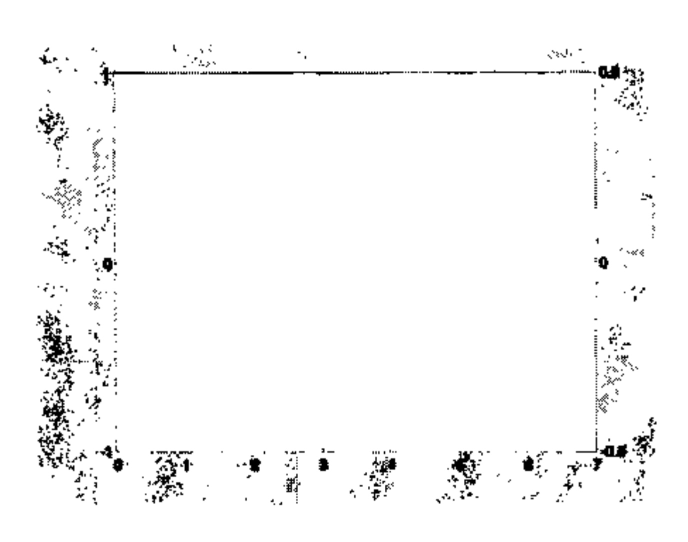


图 16-27　使用 plot 函数和两个 y 轴创建图形

7. 为当前轴添加标题

名称：title

为当前轴添加标题。

语法：该函数有如下四种表达形式：

- title('string')
- title(fname)
- title(...,'PropertyName',PropertyValue,...)
- h = title(...)

描述：每个轴类图形对象可以有一个标题。标题放置在轴的上方居中的位置处。

title('string')在当前轴的上方居中的位置输出 string 中的文本串作为标题。

title(fname)把 fname 函数返回的文本串作为标题显示在当前轴的上方居中处。

title(...,'PropertyName',PropertyValue,...)定义标题所生成的文本对象的 property name 属性的值。

h = title(...)返回作为标题的文本对象的句柄。

举例：

(1) 把当天的日期作为轴的标题。

title(date)

Include a variable's value in a title:

f = 70;

c = (f--32)/1.8;

title(['Temperature is ',num2str(c),'C'])

(2) 标题种含有一个变量并设置标题的颜色为绿色：

n = 3;

title(['Case number #',int2str(n)],'Color','y')

(3) 标题中含有希腊符号：

title('\ite^{\omega\tau} = cos(\omega\tau) + isin(\omega\tau)')

(4) 标题中含有上标符号：

title('\alpha^2')

(5) 标题中含有下标符号：

title('X_1')

8. 标注 x、y 和 z 轴

名称：xlabel, ylabel, zlabel

标注 x、y、和 z 轴。

语法：该函数有如下八种表达形式

- xlabel('string')
- xlabel(fname)
- xlabel(...,'PropertyName',PropertyValue,...)
- h = xlabel(...)
- ylabel(...)
- h = ylabel(...)
- zlabel(...)
- h = zlabel(...)

描述：每一个轴图形对象能够有一个关于 x、y 和 z 轴的标注。在二维图形中，标注出现在其对应轴的下方；在三维图形中，标注出现在其对应的轴的旁边或下方。

xlabel('string')标注当前轴的 x 轴。

xlabel(fname)估计函数的 fname，它必须返回一个字符串，然后在 x 轴的旁边显示这个字符串。

xlabel(...,'PropertName',PropertyValue,...)为 xlabel 创建的文本图形对象确定属性名和属性值。

h = xlabel(...)、 h = ylabel(...)和 h = zlabel(...)返回指向作为标注使用的文本对象的一个句柄。

ylabel(...)和 zlabel(...)分别标注当前轴的 y 轴和 z 轴。

16.4　表面、网格和轮廓绘制

1. 二维等高线图

名称：contour

二维等高线图。

语法：该函数有如下八种表达形式：

- contour(Z)
- contour(Z,n)
- contour(Z,v)
- contour(X,Y,Z)

- contour(X,Y,Z,n)
- contour(X,Y,Z,v)
- contour(...,LineSpec)
- [C,h] = contour(...)

描述：contour 命令显示矩阵 Z 的等高线。使用 clabel 可标注等高线。

contour(Z)画出矩阵 Z 的等高线图，这里 Z 是相对于 x-y 平面的高度。Z 至少是一个 2×2 矩阵。等高线的层数和等高线的层值可基于 Z 的最大和最小值自动加以选择。x 轴和 y 轴的范围是[1:n]和[1:m]，这里[m,n] = size(Z)。

contour(Z,n)画出矩阵 Z 的 n 层等高线图。

contour(Z,v)在向量 v 指定的数据值处，画出 Z 的等高线。等高线的层数等于 length(v)。使用 contour(Z,[i i])可画出层为 i 的一条单等高线。

contour(X,Y,Z)、contour(X,Y,Z,n)和 contour(X,Y,Z,v)画出 Z 的等高线。X 和 Y 确定 x 轴和 y 轴的范围。当 X 和 Y 是矩阵时，它们必须和 Z 有相同的阶数，在这种情况下，它们确定了一个表面，正如 surf 所做的那样。

contour(...,LineSpec)使用线型和由 LineSpec 指定的颜色画出等高线。等高线忽略标记。

[C,h] = contour(...)返回等高线矩阵 C 和一个指向图形对象的句柄。clabel 使用等高线矩阵 C 创建标记。contour 创建块图形句柄，除非用户指定了 LineSpec，在这种情况下，contour 创建线图形句柄。

举例：

(1) 在区间$-2\pi \le x \le 2\pi$, $-2\pi \le y \le 2\pi$上画出函数 $z = x*\exp(-x^2 - y^2)$的登高线图。

```
[X,Y] = meshgrid(-2:.2:2,-2:.2:3);
Z = X.*exp(-X.^2-Y.^2);
[C,h] = contour(X,Y,Z);
clabel(C,h)
colormap cool
```

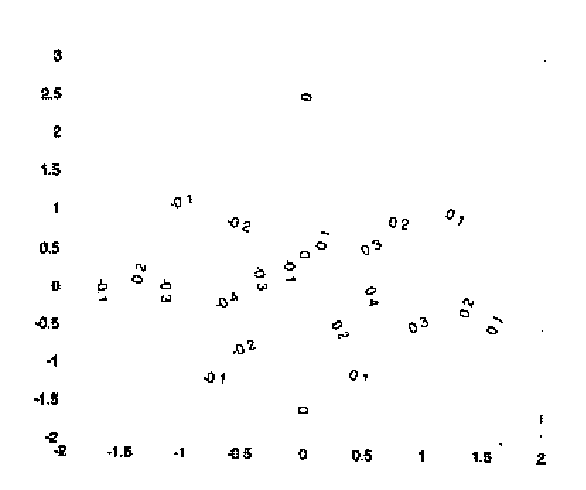


图 16-28 等高线图

(2) 用 20 条等高线画出上面的图形。

contour(X,Y,Z,20)

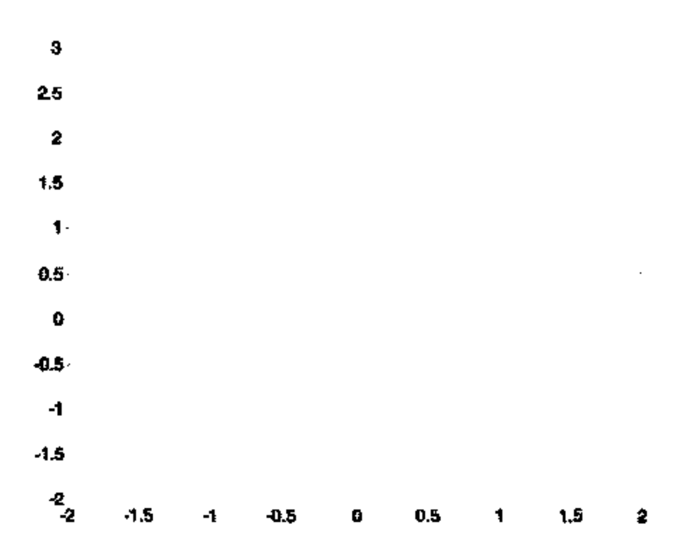


图 16-29 等高线图

(3) 使用 interp2 和 contour 创建等高线。

```
Z = magic(4);
[C,h] = contour(interp2(Z,4));
clabel(C,h)
```

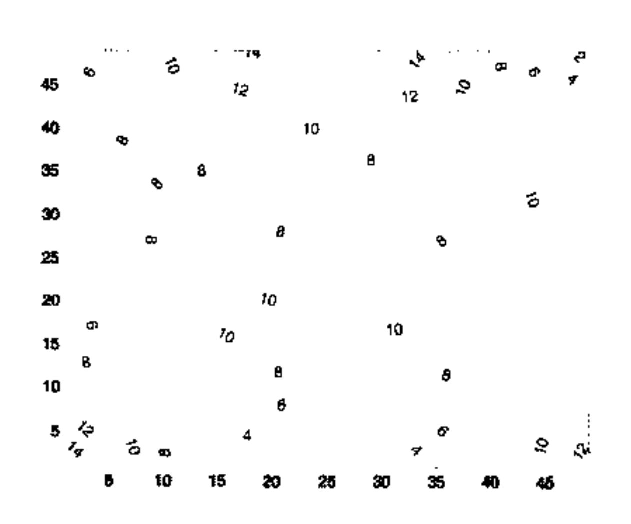


图 16-30 使用 interp2 和 contour 创建带标注的等高线

2. 低层等高线图形计算

名称：contourc

低层等高线图形计算。

语法：该函数有如下六种表达形式：

- C = contourc(Z)
- C = contourc(Z,n)

- C = contourc(Z,v)
- C = contourc(x,y,Z)
- C = contourc(x,y,Z,n)
- C = contourc(x,y,Z,v)

描述：contourc 使用 contour、contour3 和 contourf 命令计算等高线矩阵 C。Z 中的值决定了关于一个平面的等高线的高度。等高线的计算使用由 Z 的尺度确定的规则空间网格。

C = contourc(Z)根据矩阵 Z 中的数据计算等高线矩阵，这里 Z 必须至少为 2×2 矩阵。等高线在 Z 的单位长度里必须是等值线。等高线的数目和对应的等高线的值是自动选择的。

C = contourc(Z,n)用 n 个等级的等高线计算矩阵 Z 的等高线数据。

C = contourc(Z,v)用向量 v 所指定的等高线值来计算矩阵 Z 的等高线。v 的长度决定等高线等级的数目。为了计算等级为 I 的一条单个的等高线，可以使用命令 contourc(Z,[i i]).。

C = contourc(x,y,Z)、C = contourc(x,y,Z,n)和 C = contourc(x,y,Z,v)使用确定 x 轴和 y 轴界限的向量 x 和 y 计算 Z 的等高线。x 和 y 必须是单调增加的。

3. 填充二维等高线图

名称：contourf

填充二维等高线图。

语法：该函数有如下七种表达形式：

- contourf(Z)
- contourf(Z,n)
- contourf(Z,v)
- contourf(X,Y,Z)
- contourf(X,Y,Z,n)
- contourf(X,Y,Z,v)
- [C,h,CF] = contourf(...)

描述：经过填充的等高线图可以在等直线间的区域显示所填充的颜色。该颜色取决与当前图形的色图。

contourf(Z)绘制矩阵 Z 的等高线图，其中 Z 可以理解为相对一个平面的高度。Z 必须至少为一个 2×2 的矩阵。所显示的等高线的数目和大小为自动选定。

contourf(Z,n)绘制矩阵 Z 的 n 条等高线的等高线图。

contourf(Z,v)绘制的等高线图中的等高线的值由向量 v 指定。

contourf(X,Y,Z),contourf(X,Y,Z,n),和 contourf(X,Y,Z,v)在 X 和 Y 所限制的范围内绘制矩阵 Z 的等高线图。当 X 和 Y 是矩阵时，它们必须和 Z 是同阶的矩阵，此时它们定义了一个类似 surf 所生成的表面。

[C,h,CF] = contourf(...)返回 contourc 函数所生成的等高线矩阵 C，一个指向块图形对象的句柄向量 h 和一个填充色矩阵 CF。

举例：

绘制一个有填充色的 peak 函数的等高线图。

```
[C,h] = contourf(peaks(10),5);
colormap autumn
```

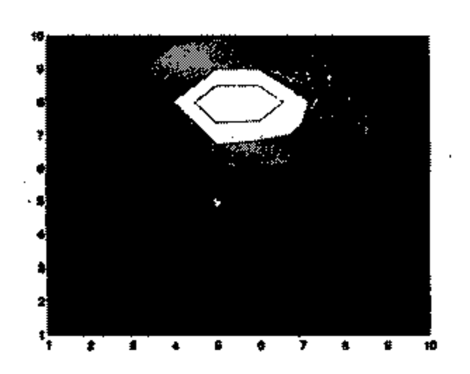


图 16-31　具有填充色的等高线图图

4. 从一个网线图中删除消隐线

名称：hidden

从一个网线图中删除消隐线。

语法：该函数有如下三种表达形式：

- hidden on
- hidden off
- hidden

描述：删除消隐线只是画出那些在视图区域内不被其他对象阻碍的线。

hidden on 为当前的图形打开删除消隐线命令，一个网线后面的线就被那些在它前面的线所隐藏。这是默认的行为。

hidden off 为当前图形关闭删除消隐线命令。

hidden 命令在删除消隐线状态之间进行切换。

举例：

在显示时 peaks 函数，设置打开或关闭消隐线删除命令。

```
mesh(peaks)
hidden off
hidden on
```

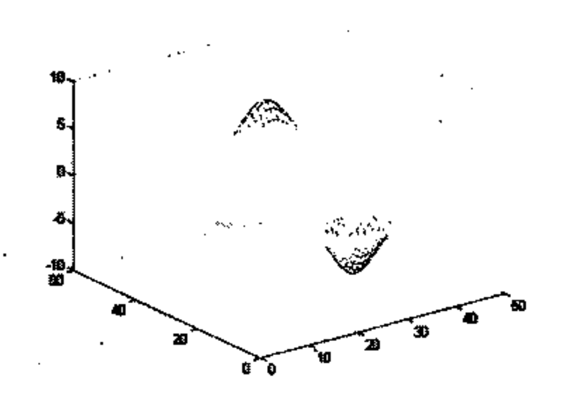


图 16-32　消隐线删除命令的影响

5. 网线图

名称：mesh、meshc、meshz

网线图。

语法：该函数有如下八种表达形式：

- mesh(X,Y,Z)
- mesh(Z)
- mesh(...,C)
- meshc(...)
- meshz(...)
- h = mesh(...)
- h = meshc(...)
- h = meshz(...)

描述：mesh、meshc、和 meshz 生成由 X，Y 和 Z 定义的网线图，其颜色由 C 定义。

mesh(X,Y,Z)绘制由 Z 定义颜色的网线图，因此颜色和曲面的高度成正比。如果 X 和 Y 是向量，length(X) = n、length(Y) = m，且[m,n] = size(Z)，则(X(j), Y(i), Z(i, j))是网线的交叉点；如果 X 和 Y 是矩阵，则(X(i, j), Y(i, j), Z(i, j))是交叉点。

mesh(Z)生成的网线图满足 X = 1:n、Y = 1:m，其中[m,n] = size(Z)，而高度 Z 是定义在矩形区域上的单值函数，网线的颜色与表面的高度成正比。

mesh(...,C)绘制的网线图的颜色由矩阵 C 定义。MATLAB 从当前的色图到 C 中的数据执行一个线性变换以得到网线的颜色。如果 X，Y 和 Z 是矩阵，则它们的阶数必须相同。

meshc(...) 在网线图的下面绘制等高线图。

meshz(...)在网线图的周围绘制窗帘图。

h = mesh(...)，h = meshc(...)和 h = meshz(...)返回指向曲面对象的句柄。

举例：

(1) 生成 peaks 曲面的一个网线图和等高线图的混合图。

```
[X,Y] = meshgrid(-3:.15:3);
Z = peaks(X,Y);
meshc(X,Y,Z);
axis([-3 3 -3 3 -10 5])
```

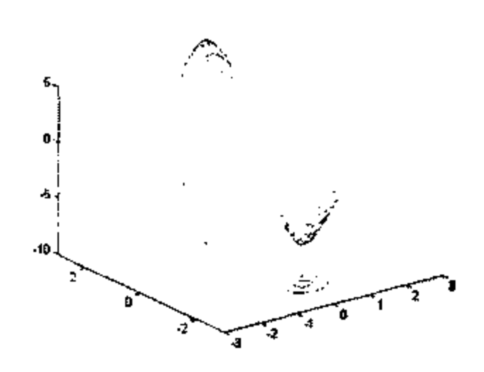


图 16-33　网线图和等高线图的混合图

(2) 生成 peaks 函数的窗帘图。

```
[X,Y] = meshgrid(-3:.15:3);
Z = peaks(X,Y);
meshz(X,Y,Z)
```

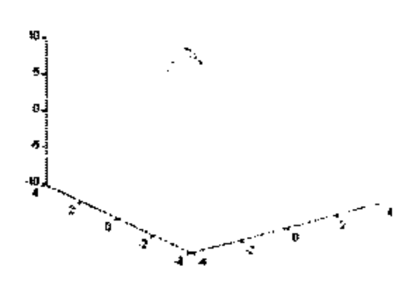


图 16-34 窗帘图

6. 两变量的样本函数

名称：peaks

两变量的样本函数。

语法：该函数有如下十一种表达形式：

- Z = peaks;
- Z = peaks(n);
- Z = peaks(V);
- Z = peaks(X,Y);
- peaks;
- peaks(N);
- peaks(V);
- peaks(X,Y);
- [X,Y,Z] = peaks;
- [X,Y,Z] = peaks(n);
- [X,Y,Z] = peaks(V);

描述：peaks 是一个两变量的函数，由 Gaussian 分布转化和缩放得到，帮助演示 mesh，surf，pcolor 和 contour 等函数。

Z = peaks 返回一个 49×49 的矩阵。Z = peaks(n)返回一个 n×n 的矩阵。Z = peaks(V) 返回一个 n×n 的矩阵，其中 n = length(V)。Z = peaks(X,Y)在给定的 X 和 Y（两者必须同阶）处估算 pesks，返回一个同阶的矩阵。无参数的 peaks(...)利用 surf 函数绘制 peaks 函数。

[X,Y,Z] = peaks(...)返回两个附加矩阵 X 和 Y，作为参数为如 surf(X,Y,Z,del2(Z))利用。如果不是作为输入量，则隐含的矩阵 X 和 Y 是：[X,Y] = meshgrid(V,V)，其中 V 是一个给定的向量，或者 V 是一个长度为 n 的向量，它的元素等分-3 到 3 的区间。如果没有给定参数，则默认的 n 值是 49。

7. 三维阴影表面图

名称：surf, surfc

三维阴影表面图。

语法：该函数有如下七种表达形式：

- surf(Z)
- surf(X,Y,Z)
- surf(X,Y,Z,C)
- surf(...,'PropertyName',PropertyValue)
- surfc(...)
- h = surf(...)
- h = surfc(...)

描述：使用 surf 和 surfc 可以观看一个矩形区域上的数学函数图。surf 和 surfc 创建由 X、Y 和 Z 以及颜色所确定的着色参数表面，其中颜色是由 Z 或 C 确定的。

使用 x = 1:n 和 y = 1:m，surf(Z)根据矩阵 Z 中的 z 分量创建一个三维着色表面，这里[m,n] = size(Z)。高度 Z 是一个定义在矩形网格上的单值函数。Z 在确定表面高度的同时，还确定了颜色数据，因此颜色和表面高度成比例。

surf(X,Y,Z)使用 Z 作为高度和颜色数据创建一个着色表面。X 和 Y 是定义表面的 x 和 y 分量的向量或矩阵。如果 X 和 Y 是向量，则 length(X) = n 和 length(Y) = m，这里[m,n] = size(Z)。在这种情况下，表面向量是(X(j),Y(i)Z(i,j))。

surf(X,Y,Z,C)使用由 C 定义的颜色创建一个着色表面。为了从当前色图中得到颜色，MATLAB 对这个数据执行一个线性变换。

surf(...,'PropertyName',PropertyValue)随同数据一起确定表面属性。

surfc(...)在表面下方画出等高线图。

h = surf(...) and h = surfc(...)返回指向一个表面图形对象的句柄。

举例：

(1) 显示 peaks 面的表面和等高线图形。

```
[X,Y,Z] = peaks(40);
surfc(X,Y,Z)
colormap hsv
axis([-4 4 -4 4 -10 5])
```

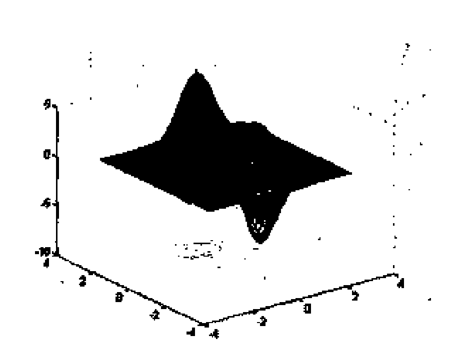


图 16-35 peaks 的表面图和等高线图

(2) 使用 Hadamard 矩阵的+1 和-1 模式为球面着色。

```
k = 5;
n = 2^k-1;
[x,y,z] = sphere(n);
c = hadamard(2^k);
surf(x,y,z,c);
colormap([1   1   0; 0   1   1])
axis equal
```

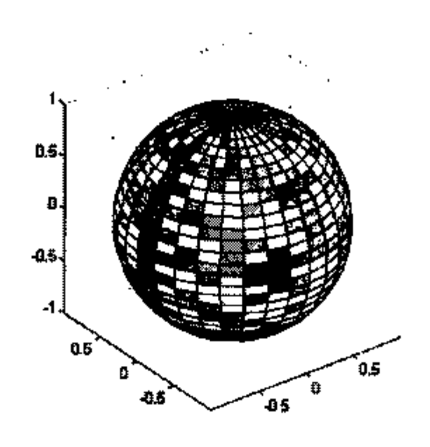


图 16-36 球面的着色图

8. 带有基于色图照明的表面图

名称：surfl

带有基于色图照明的表面图。

语法：该函数有如下六种表达形式：

- surfl(Z)
- surfl(X,Y,Z)
- surfl(...,'light')
- surfl(...,s)
- surfl(X,Y,Z,s,k)
- h = surfl(...)

描述：surfl 函数显示一个基于背景光，漫射光和定向光的综合的光模式处理过的阴影图。surfl(Z)和 surfl(X,Y,Z)画出基于默认的光源光照方向，阴影模式的光照系数的三维阴影图。X、Y 和 Z 是向量或矩阵，定义了曲面的 x，y 和 z 分量。

surfl(...,'light')利用 MATLAB 的光照对象的定义生成一个彩色的代光照的表面图。不同的默认光照模式其效果也不同。surfl(...,'cdata')改变表面的反射光的颜色。

surfl(...,s)指定光源的方向。s 是一个二维或三维的向量，定义了一个表面到一个光源的方向。s = [sx sy sz]或 s = [azimuth elevation]。默认值是当前观测方向逆时针旋转 45°。

surfl(X,Y,Z,s,k)指定了反射比常量。k 是一个四元素的向量，定义了背景光、漫射光、

定向光的份额和扩散系数。k = [ka kd ks shine]，其默认值是[.55,.6,.4,10]。

h = surfl(...)返回一个指向曲面图形对象的句柄。

举例:

(1) peaks 函数在控制光照下的三维表面。

```
[x,y] = meshgrid(-3:1/6:3);
z = peaks(x,y);
surfl(x,y,z);
shading interp
colormap(gray);
axis([-3  3  -3  3  -8  8])
```

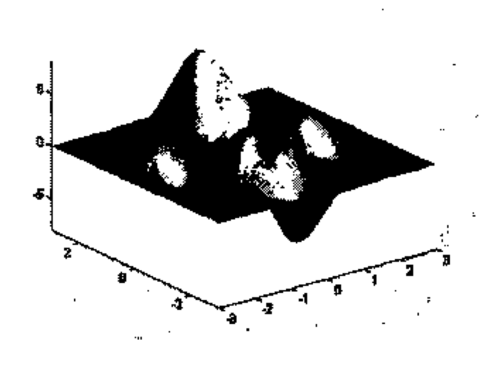


图 16-37　光照下的三维表面图

(2) 非默认值条件下观测 peaks 函数的三维表面。

```
view([10 10])
grid on
hold on
surfl(peaks)
shading interp
colormap copper
hold off
```

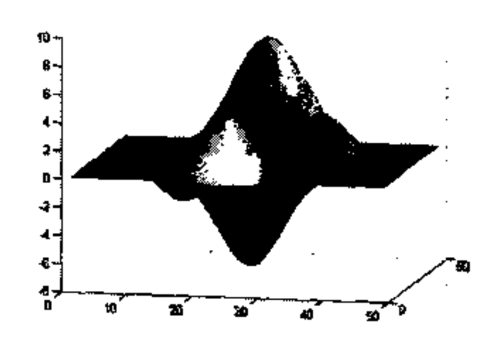


图 16-38　非默认条件下函数的三维表面

9. 三角形网线图

名称：trimesh

三角形网线图。

语法：该函数有如下四种表达形式：

- trimesh(Tri,X,Y,Z)
- trimesh(Tri,X,Y,Z,C)
- trimesh(...'PropertyName',PropertyValue...)
- h = trimesh(...)

描述：trimesh(Tri,X,Y,Z)显示的是作为网线定义在 m×3 矩阵 Tri 里的三角网线。通过将索引编入包含 X、Y 和 Z 顶点的向量或矩阵，Tri 的每一行定义了一个单个的三角面。

trimesh(Tri,X,Y,Z,C)用和 surf 函数同样的方式确定由 C 定义的颜色。为了从当前色图中获得颜色，MATLAB 在这个数据上执行一个线性变换。

trimesh(...'PropertyName',PropertyValue...)确定了附加的块属性名字和值，块属性值是关于由函数创建的块图形对象的。

h = trimesh(...)返回一个指向块图形对象的句柄。

举例：

(1) 创建顶点向量和面矩阵，并创建一个三角形网线图。

```
x = rand(1,100);
y = rand(1,100);
z = peaks(10*x-3,10*x-3);
tri = delaunay(x,y);
trimesh(tri,x,y,z)
```

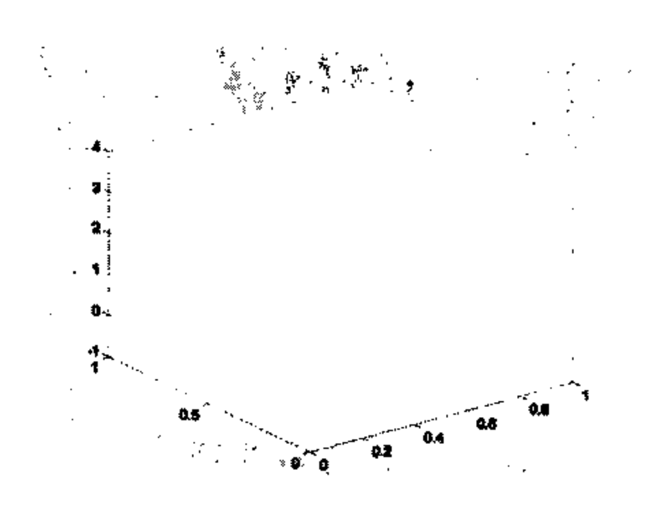


图 16-39 三角形网线图

10. 三角形表面图

名称：trisurf

三角形表面图。

语法：该函数有如下四种表达形式：

- trisurf(Tri,X,Y,Z)
- trisurf(Tri,X,Y,Z,C)
- trisurf(...'PropertyName',PropertyValue...)
- h = trisurf(...)

描述：trisurf(Tri,X,Y,Z)显示 m×3 的面矩阵 Tri 中定义的的三角形。Tri 的每一行定义一个三角形平面。

trisurf(Tri,X,Y,Z,C)按照 C 定义的颜色同 surf 函数一样绘制表面图。MATLAB 从当前 colormap 中通过线性变换得到颜色定义。

trisurf(...'PropertyName',PropertyValue...)为块图形对象指定额外的块属性名称及其数值。

h = trisurf(...)返回一个块句柄。

举例：

绘制一个三角形表面图。

```
x = rand(1,20);
y = rand(1,20);
z = peaks(3*x-3,3*x-3);
tri = delaunay(x,y);
trisurf(tri,x,y,z)
```

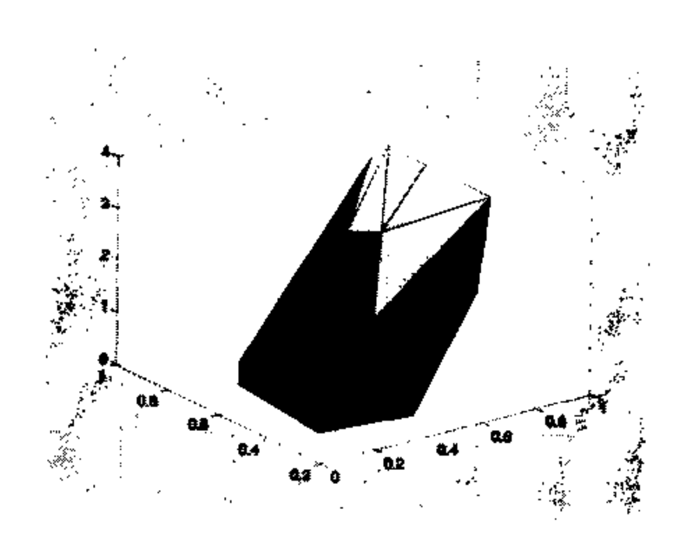


图 16-40　三角形表面图

16.5　体数据可视化

1. 三维向量场中的速度向量锥图

名称：coneplot

在三维向量场中将速度向量画成锥形。

语法：该函数有如下六种表达形式：

- coneplot(X,Y,Z,U,V,W,Cx,Cy,Cz)
- coneplot(U,V,W,Cx,Cy,Cz)

- coneplot(...,s)
- coneplot(...,'quiver')
- coneplot(...,'method')
- h = coneplot(...)

描述：coneplot(X,Y,Z,U,V,W,Cx,Cy,Cz)绘出一个锥形速度向量图，方向是指向速度向量的方向，长度是与速度向量的大小成比例的：X、Y、Z 定义关于速度场的坐标；U、V、W 定义速度场。这些数组必须具有相同的维数，必须是单调且呈现三维网格状（例如由 meshgrid 产生的数据）；Cx、Cy、Cz 定义向量场中锥的位置。

coneplot(U,V,W,Cx,Cy,Cz)(忽略 X、Y 和 Z 分量)假设[X,Y,Z] = meshgrid(1:n,1:m,1:p)，这里[m,n,p] = size(U)。

coneplot(...,s)MATLAB 自动调整锥的尺度，然后通过尺度因子 s 来拉长它们。如果用户不想给 s 赋予一个确定的值，MATLAB 就规定 s 的值为 1，如果要画出没有自动调节功能的锥图，用户可以使用 s = 0 的设置。

coneplot(...,'quiver')画出的是箭头，而不是锥。

coneplot(...,'method')确定所用的插值方法。插值方法有线性插值、立方插值，最近插值，其中线性插值是默认的。

h = coneplot(...)返回一个指向用于绘制锥图的块对象的句柄。用户可以使用 set 命令来改变锥图的属性。

举例：

本例为一个表示空气运动的向量体数据画出速度向量锥。最终的图形使用了大量的增强功能使数据更有效的可视化。它们包括：锥图指明了风速的量值和方向；放置在数据范围两端的切片平面为长方体里的锥图提供了可视化的环境；定向的照明效果提供了关于锥的方向的可视化咨询；通过选择视点，投影类型和量值，视图调节组成了最能揭示数据信息内容的上下文。

```
load wind
xmin = min(x(:));
xmax = max(x(:));
ymin = min(y(:));
ymax = max(y(:));
zmin = min(z(:));
daspect([2,2,1]);
xrange = linspace(xmin,xmax,8);
yrange = linspace(ymin,ymax,8);
zrange = 3:4:15;
[cx cy cz] = meshgrid(xrange,yrange,zrange);
hcones = coneplot(x,y,z,u,v,w,cx,cy,cz,5);
set(hcones,'FaceColor','red','EdgeColor','none')
hold on;
wind_speed = sqrt(u.^2 + v.^2 + w.^2);
```

```
hsurfaces = slice(x,y,z,wind_speed,[xmin,xmax],ymax,zmin);
set(hsurfaces,'FaceColor','interp','EdgeColor','none')
hold off
axis tight; view(30,40); axis off
camproj perspective; camzoom(1.5)
camlight right; lighting phong
set(hsurfaces,'AmbientStrength',.6)
set(hcones,'DiffuseStrength',.8)
```

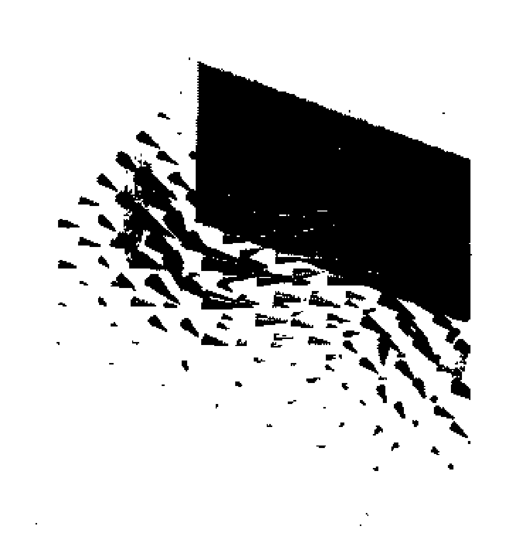


图 16-41　速度场图

2. 在三维物体切面上绘制等高线

名称：contourslice

在三维物体切面上绘制等高线。

语法：该函数有如下八种表达形式：

- contourslice(X,Y,Z,V,Sx,Sy,Sz)
- contourslice(X,Y,Z,V,Xi,Yi,Zi)
- contourslice(V,Sx,Sy,Sz), contourslice(V,Xi,Yi,Zi)
- contourslice(...,n)
- contourslice(...,cvals)
- contourslice(...,[cv cv])
- contourslice(...,'method')
- h = contourslice(...)

描述：contourslice(X,Y,Z,V,Sx,Sy,Sz)在 x、y 和 z 坐标系里，在由向量 Sx、Sy、Sz 确定的平面上画出等高线。数组 X、Y 和 Z 为块体 V 定义了坐标系，它们必须是单调的和三维网格状的。每一个等高线的颜色由块体 V 决定，必须是一个 m×n×p 的体数组。

contourslice(X,Y,Z,V,Xi,Yi,Zi)在块体 V 里，沿着由数组 Xi、Yi、Zi 定义的面绘制等高线。

contourslice(V,Sx,Sy,Sz)和 contourslice(V,Xi,Yi,Zi)忽略 X、Y 和 Z 选项的两个命令假设[X,Y,Z] = meshgrid(1:n,1:m,1:p)，这里[m,n,p]= size(v)。

contourslice(...,n)在每个平面上画出 n 条等高线，忽略默认值。

contourslice(...,cvals)在每个由向量 cvals 定义的平面上绘制 length(cval)条等高线。

contourslice(...,[cv cv])在层为 cv 的每个平面上计算一条单个等高线。

contourslice(...,'method')指定了插值方法。插值方法有线性插值、立方插值和最近插值。除了当等高线是沿 Xi、Yi、Zi 定义的表面画出的(在这种情况下线性插值是默认的)外，最近插值是默认的。

h = contourslice(...)返回指向用于实现等高线的块对象的句柄向量。

举例：

本例使用流体数据来阐明等高线切片平面的用处。

为 Sx 指定一个 9 维向量，为 Sy 指定一个空向量，为 Sz 指定一个值为 0 的标量。它在 y-z 平面，沿 x 方向创建了 9 个等高线图，在 z = 0 的 x-y 平面创建一个等高线图。使用 linspace 来定义一个从-8 到 2 的十元素等间距向量，它在每一个间隔处指定了等高线的数目。定义视图和投影类型(camva, camproj, campos)。设置图形和轴的类型。

```
[x y z v] = flow;
h = contourslice(x,y,z,v,[1:9],[],[0],linspace(-8,2,10));
axis([0,10,-3,3,-3,3]); daspect([1,1,1])
camva(24); camproj perspective;
campos([-3,-15,5])
set(gcf,'Color',[.5,.5,.5],'Renderer','zbuffer')
set(gca,'Color','black','XColor','white', ...
     'YColor','white','ZColor','white')
box on
```



图 16-42　等高线切片平面图

3. 计算帽端等表面几何

名称：isocaps

计算帽端等表面几何。

语法：该函数有如下六种表达形式：

- fvc = isocaps(X,Y,Z,V,isovalue)
- fvc = isocaps(V,isovalue)
- fvc = isocaps(...,'enclose')
- fvc = isocaps(...,'whichplane')
- [f,v,c] = isocaps(...)
- isocaps(...)

描述：fvc = isocaps(X,Y,Z,V,isovalue)为在等表面值 isovalue 处的块体数据 V 计算帽端等表面几何。数组 X、Y 和 Z 定义了关于块体 V 的坐标系。

数据结构 fvc 包含了关于帽端的面，顶点和颜色数据，并能直接通向 patch 命令。

fvc = isocaps(V,isovalue)假设 X、Y 和 Z 被定义为[X,Y,Z] = meshgrid(1:n,1:m,1:p)，这里，[m,n,p] = size(V)。

fvc = isocaps(...,'enclose')确定帽端的环绕数据值'enclose'是否在 isovalue 值之上或之下。字符串 enclose 不是在它之上(默认)，就是在它之下。

fvc = isocaps(...,'whichplane')确定的是在哪个平面上画出帽端。关于在哪个平面的可能的值是：all (default)，xmin，xmax，ymin，ymax，zmin 或 zmax。

[f,v,c] = isocaps(...)返回关于帽端而不是关于数据结构 fve 的面、顶点和颜色。

没有输出项的 isocaps(...)用计算过的面、顶点和颜色画出一个块。

举例：

本例使用一个人类头骨的 MRI 切片集合作为一个数据集合。它阐明了 isocaps 的用处，并画出切割后的帽端。

红色的 isosurface 显示了块体(头骨)的概貌，帽端显示了块体的内部；从帽端数据(p2)创建的 patch 使用了插值的面颜色，这意味着灰色的色图和光源决定它是如何着色的。isosurface patch (p1)使用了一个由照明影响的浅红色的面颜色，但没有使用色图。

```
load mri
D = squeeze(D);
D(:,1:60,:) = [];
p1 = patch(isosurface(D, 5),'FaceColor','red',...
     'EdgeColor','none');
p2 = patch(isocaps(D, 5),'FaceColor','interp',...
     'EdgeColor','none');
view(3); axis tight; daspect([1,1,.4])
colormap(gray(100))
camlight left; camlight; lighting gouraud
isonormals(D,p1)
```

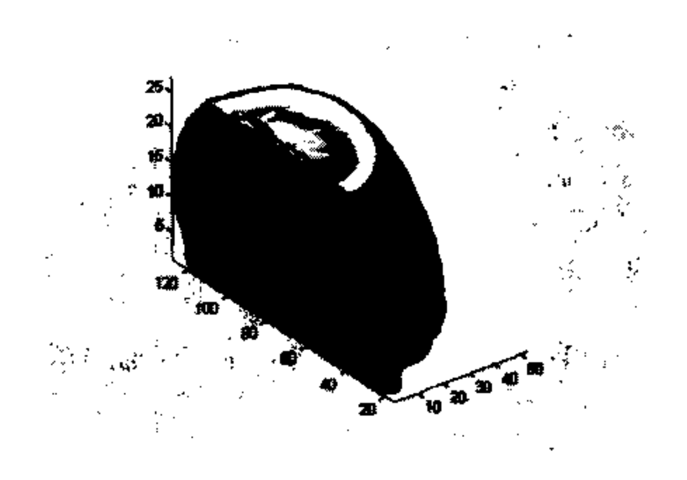


图 16-43　人类头骨的帽端图

4. 计算等值表面顶点的法向

名称：isonormals

计算等值表面顶点的法向。

语法：该函数有如下五种表达形式：

- n = isonormals(X,Y,Z,V,vertices)
- n = isonormals(V,vertices)
- n = isonormals(V,p), n = isonormals(X,Y,Z,V,p)
- n = isonormals(...,'negate')
- isonormals(V,p), isonormals(X,Y,Z,V,p)

描述：n = isonormals(X,Y,Z,V,vertices)从顶点列表中的顶点，使用数据 V 的梯度，来计算等值表面顶点的法向。数组 X、Y 和 Z 定义了块体 V 的坐标系。所计算的法向返回到 n 里。

n = isonormals(V,vertices)假设数组 X、Y 和 Z 被定义为[X,Y,Z] = meshgrid(1:n,1:m,1:p)，这里[m,n,p] = size(V)。

n = isonormals(V,p)和 n = isonormals(X,Y,Z,V,p)从由句柄 p 确定的块的顶点来计算法向。n = isonormals(...,'negate')取消法向。

isonormals(V,p)和 isonormals(X,Y,Z,V,p)将由句柄 p 确定的块的 VertexNormals 属性设置为已计算过的法向而不是返回其值。

举例：

本例比较了不同的表面法向。在这种情况下，用于画出等值面的 triangles 定义了法向。在其它情况下，isonormals 函数使用块体数据来计算基于数据点梯度的顶点法向。后一种方法逐步产生一个较为光滑的等值面。

定义一个体数据的三维数组：

```
data = cat(3, [0 .3 0; 0 .5 0; 0 0 0], ...
              [.1 .6 0; 0 2 0; .4 .6 0],...
              [3 .2 .1; .1 5 0;.2 .3 0]);
data = interp3(data,3,'cubic');
```

从块体数据和增加照明来画出一个等值表面。这个 isosurfacee 使用 triangle 法向向量(patch, isosurface, view, daspect, axis, camlight, lighting, title)：

```
subplot(1,2,1)
p1 = patch(isosurface(data,.5),...
'FaceColor','red','EdgeColor','none');
view(3); daspect([1,1,1]); axis tight
camlight; camlight(-80,-10); lighting phong;
title('Triangle Normals')
```

在相同照明情况下，使用从体数据中计算的 normals 来绘制等值面图：

```
subplot(1,2,2)
p2 = patch(isosurface(data,.5),...
    'FaceColor','red','EdgeColor','none');
isonormals(data,p2)
view(3); daspect([1 1 1]); axis tight
camlight;   camlight(-80,-10); lighting phong;
title('Data Normals')
```

这些等值表面阐明了 triangle 和数据法向之间的不同之处：

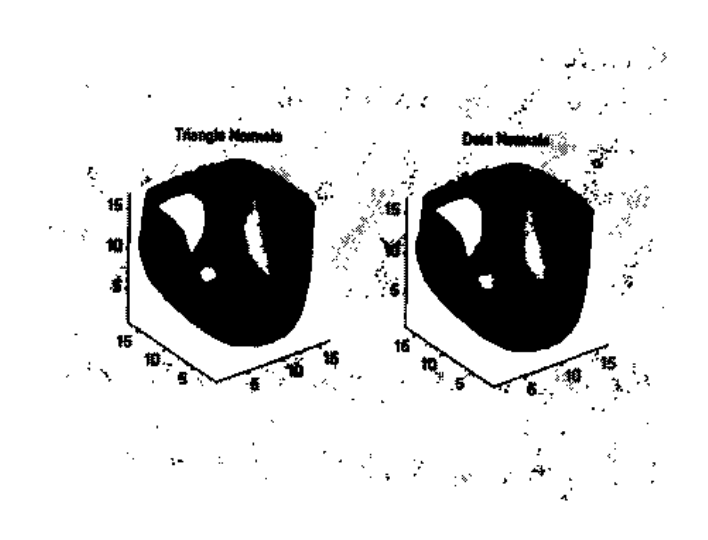


图 16-44　等法向图

5. 从块体数据中提取等表面数据

名称：isosurface

从块体数据中提取等表面数据。

语法：该函数有如下七种表达形式：

- fv = isosurface(X,Y,Z,V,isovalue)
- fv = isosurface(V,isovalue)
- fv = isosurface(X,Y,Z,V), fv = isosurface(X,Y,Z,V)
- fv = isosurface(...,'noshare')
- fv = isosurface(...,'verbose')

- [f,v] = isosurface(...)
- isosurface(...)

描述：fv = isosurface(X,Y,Z,V,isovalue)在由 isovalue 确定的等表面值处，从块体数据 V 中计算等表面数据。数组 X、Y 和 Z 为块体 V 定义了坐标系。数据结构 fv 包含了等表面的面和顶点，用户可以直接通向 patch 命令。

fv = isosurface(V,isovalue)假设数组 X、Y 和 Z 被定义为[X,Y,Z] = meshgrid(1:n,1:m,1:p)这里，[m,n,p] = size(V)。

fv = isosurface(...,'noshare')不创建共享顶点。这样可以快速产生较大的一组顶点。

随着计算的进行，fv = isosurface(...,'verbose')向命令窗口打印进程信息。

[f,v] = isosurface(...)用两个数组返回面和顶点，而不是用一个数据结构返回面和顶点。

没有输出项的 isosurface(...)使用计算过的面和顶点创建一个块。

举例：

本例使用流动数据集，它表示在一个无限大的槽里的一个沉入水中的喷气机(a submerged jet)的速度剖面。等表面在值为-3 处画出。在 patch 命令之后的说明通过下面为等表面准备照明：(1)根据块体数据重新计算等表面的法向(isonormals)；(2)设置面和边的颜色(set, FaceColor, EdgeColor)；(3)指定视图(daspect, view)；(4)增加照明(camlight, lighting)。

```
[x,y,z,v] = flow;
p = patch(isosurface(x,y,z,v,-3));
isonormals(x,y,z,v,p)
set(p,'FaceColor','red','EdgeColor','none');
daspect([1 1 1])
view(3)
camlight
lighting phong
```

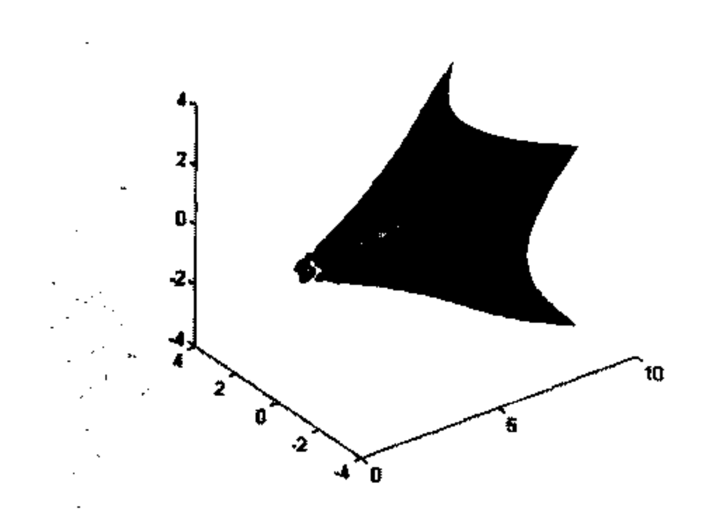


图 16-45　速度剖面的等表面图

6. 缩减块体表面的数目

名称：reducepatch

缩减块体表面的数目。

语法：该函数有如下种五表达形式：

- reducepatch(p,r)
- nfv = reducepatch(p,r)
- nfv = reducepatch(fv,r)
- nfv = reducepatch(f,v,r)
- [nf,nv] = reducepatch(...)

描述：reducepatch(p,r)减少由句柄 p 确定的块的表面的数目，同时设法保持原物体的形状。MATLAB 按照两种方式处理缩减系数 r：

如果 r 小于 1，则 r 是面的初始数目的一个函数。例如，如果用户定义 r 为 0.2，则面的数目减少到初始块里面的数目的 20%

如果 r 大于或等于 1，则 r 为面的缩减后的数目。如果用户将 r 指定为 400，则面的数目将减少到 400 个面为止。

nfv = reducepatch(p,r)返回一组缩减过的面和顶点，但是没有设置块 p 的 Face 属性和 Vertices 属性。在缩减后，数据结构 nfv 包含面和顶点。

nfv = reducepatch(fv,r)在数据结构 fv 指定的面和顶点上执行缩减。

nfv = reducepatch(p)或 nfv = reducepatch(fv)使用 0.5 的缩减系数。

nfv = reducepatch(f,v,r)在 f 指定的面上和在 v 指定的顶点上执行缩减。

[nf,nv] = reducepatch(...)在数组 nf 和 nv 里返回面和顶点。

举例：

本例图解表面数目缩减到只有初始值 15%的缩减效果：

```
[x,y,z,v] = flow;
p = patch(isosurface(x,y,z,v,-3));
set(p,'facecolor','w','EdgeColor','b');
daspect([1,1,1])
view(3)
figure;
h = axes;
p2 = copyobj(p,h);
reducepatch(p2,0.15)
daspect([1,1,1])
view(3)
```

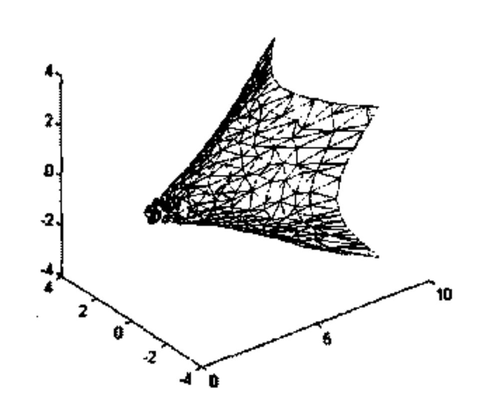


图 16-46　减缩块图

7. 缩减块体数据集中元素的数目

名称：reducevolume

缩减块体数据集中元素的数目。

语法：该函数有如下三种表达形式：

- [nx,ny,nz,nv] = reducevolume(X,Y,Z,V,[Rx,Ry,Rz])
- [nx,ny,nz,nv] = reducevolume(V,[Rx,Ry,Rz])
- nv = reducevolume(...)

描述：[nx,ny,nz,nv] = reducevolume(X,Y,Z,V,[Rx,Ry,Rz])通过在 x 方向上保留每一个第 Rx 元素，在 y 方向上保留每一个第 Ry 元素、在 z 方向上保留每一个第 Rz 元素的方式来缩减块体中的元素数目。如果一个标量 R 被用于指明其总数或减少的数目，而不是指明一个三元素向量，则 MATLAB 就假设减少为[R R R]。

数组 X、Y 和 Z 定义了块体 V 的坐标系。减缩的块体被返回在 nv 内并且其坐标系返回到 nx、ny 和 nz 内。

[nx,ny,nz,nv] = reducevolume(V,[Rx,Ry,Rz]) 假设 X、Y 和 Z 被定义为[X,Y,Z] = meshgrid(1:n,1:m,1:p)，这里[m,n,p] = size(V)。

nv = reducevolume(...)只返回减缩的块体。

举例：

本例使用了一块人的头骨的 MRI 切片作为数据集。具体步骤为：

(1) 四维数组被压缩为三维并且被缩减使得其剩余部分为每一个第 4 元素在 x 和 y 方向里，而其每一个元素在 z 方向里；

(2) 缩减的数据是光滑的(smooth3)；

(3) 头骨的概貌是一个等表面，它被产生为一个 patch (p1)，当加上照明时，其顶点法向被重新计算用来增强照明效果；

(4) 带有一个插值面颜色的第二个 patch (p2)画出帽端(FaceColor, isocaps)；

(5) 设置对象的视图(view, axis, daspect)；

(6) 一个有 100 元素的灰度刻度色图为帽端提供了颜色；

(7) 向照相机的右侧添加一个照明就显示了这个对象(camlight, lighting)。

```
load mri
D = squeeze(D);
[x,y,z,D] = reducevolume(D,[4,4,1]);
D = smooth3(D);
p1 = patch(isosurface(x,y,z,D, 5,'verbose'),...
'FaceColor','red','EdgeColor','none');
isonormals(x,y,z,D,p1);
p2 = patch(isocaps(x,y,z,D, 5),...
'FaceColor','interp','EdgeColor','none');
view(3); axis tight; daspect([1,1,.4])
colormap(gray(100))
camlight; lighting gouraud
```

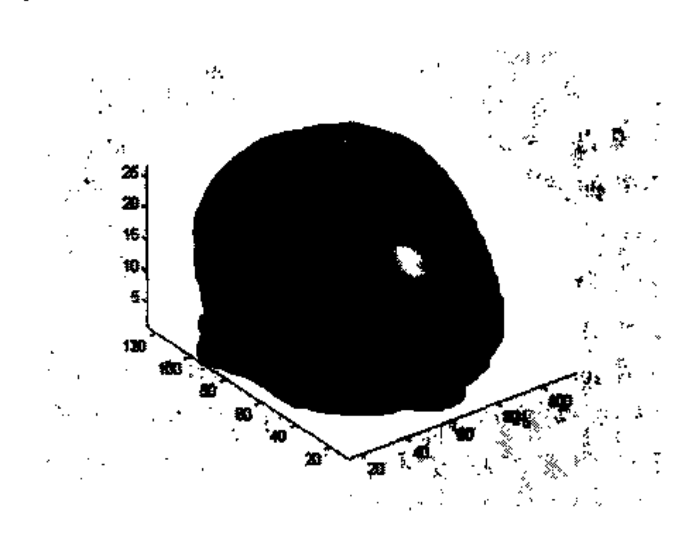


图 16-47　减缩体图

8. 缩减块体表面的尺寸

名称：shrinkfaces

缩减块体表面的尺寸。

语法：该函数有如下六种表达形式：

- shrinkfaces(p,sf)
- nfv = shrinkfaces(p,sf)
- nfv = shrinkfaces(fv,sf)
- shrinkfaces(p), shrinkfaces(fv)
- nfv = shrinkfaces(f,v,sf)
- [nf,nv] = shrinkfaces(...)

描述：shrinkfaces(p,sf)在块 p 中按照缩减系数 sf 缩减面的面积。一个等于 0.6 的缩减系数对每一个面缩减其到初始面积的 60%。如果这个块包含公共的顶点，MATLAB 在执行面积缩减前，创建非公共顶点。

nfv = shrinkfaces(p,sf)在数据结构 nfv 里返回面数据和顶点数据，但是没有设置块 p 的

Faces 和 Vertices 属性。

nfv = shrinkfaces(fv,sf)从对数据结构 fv 定义的面数据和顶点数据进行缩减。

shrinkfaces(p) and shrinkfaces(fv) (没有指定缩减系数)假设缩减系数为 0.3。

nfv = shrinkfaces(f,v,sf)对数组 f 和 v 中定义的面数据和顶点数据进行缩减。

[nf,nv] = shrinkfaces(...)返回面数据和顶点数据至两个不同的数组中，而不是返回到一个数据结构里。

举例：

下例图解了缩减前后的效果对比：

```
[x,y,z,v] = flow;
[x,y,z,v] = reducevolume(x,y,z,v,2);
fv = isosurface(x,y,z,v,-3);
p1 = patch(fv);
set(p1,'FaceColor','red','EdgeColor',[.5,.5,.5]);
daspect([1 1 1]); view(3); axis tight
title('Original')
figure
p2 = patch(shrinkfaces(fv,.3));
set(p2,'FaceColor','red','EdgeColor',[.5,.5,.5]);
daspect([1 1 1]); view(3); axis tight
title('After Shrinking')
```

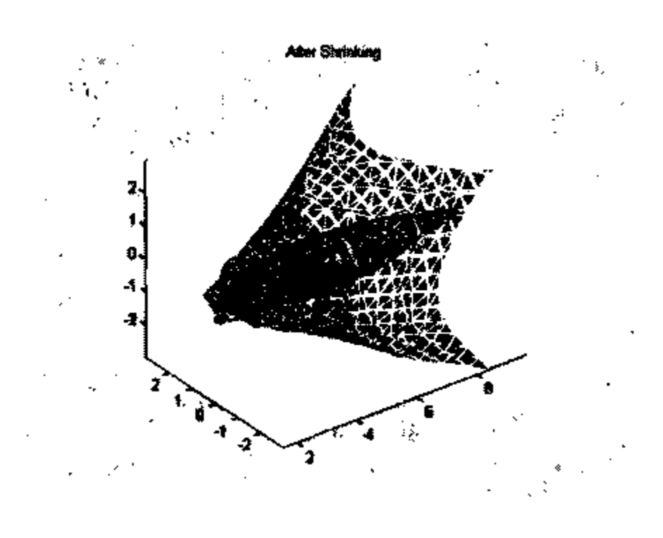


图 16-48　减缩面图

9. 三维数据光滑化

名称：smooth3

使三维数据光滑化。

语法：该函数有如下四种表达式：

- W = smooth3(V)
- W = smooth3(V,'filter')

- W = smooth3(V,'filter',size)
- W = smooth3(V,'filter',size,sd)

描述：W = smooth3(V)使输入数据 V 光滑化并且将这个光滑化后的数据返回到 W 里。

W = smooth3(V,'filter')filter 确定卷核(convolution kernel)，其可以为字符串 gaussian 或 box(默认)。

W = smooth3(V,'filter',size)设置卷核的尺寸(默认是[3 3 3])。如果尺寸是标量，它被解释为[size, size, size]。

W = smooth3(V,'filter',size,sd)设置卷核的属性。当 filter 是 gaussian 时，sd 就是标准差(默认是.65)。

举例：

本例使某个随机三维数据光滑化并用创建带有帽端的等表面。

```
data = rand(10,10,10);
data = smooth3(data,'box',5);
p1 = patch(isosurface(data,.5), ...
    'FaceColor','blue','EdgeColor','none');
p2 = patch(isocaps(data,.5), ...
      'FaceColor','interp','EdgeColor','none');
isonormals(data,p1)
view(3); axis vis3d tight
camlight; lighting phong
```

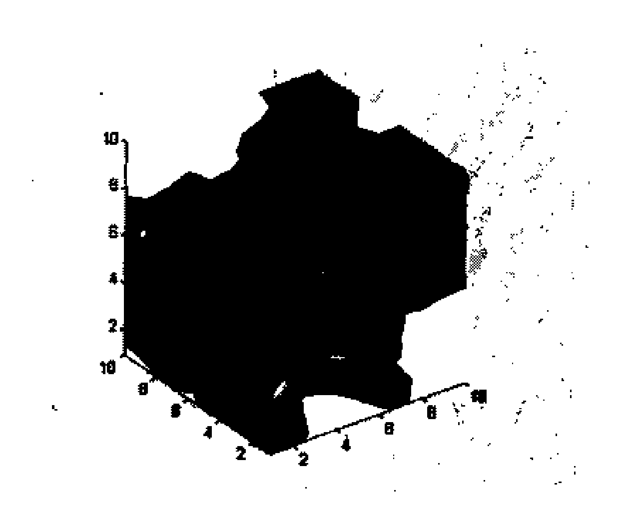


图 16-49　三维数据光滑化图

10。计算二维流线数据

名称：stream2

计算二维流线数据。

语法：该函数有如下三种表达形式：

- XY = stream2(x,y,u,v,startx,starty)
- XY = stream2(u,v,startx,starty)

- XY = stream2(...,options)

描述：XY = stream2(x,y,u,v,startx,starty)从向量数据 u 和 v 中计算流线。数组 x 和 y 为 u 和 v 定义了坐标系，并且它们必须是单调的和二维网格状的。startx 和 starty 定义了流线的起始位置。返回值 XY 包含一个顶点数组的单元数组。

XY = stream2(u,v,startx,starty)假设数组 x 和 y 被定义为[x,y] = meshgrid(1:n,1:m)，这里[m,n] = size(u)。

当创建流线时，XY = stream2(...,options)确定了使用过的选项。可以定义 options 为一个单元素向量或二元素向量，它包含流线内顶点的阶梯尺寸和顶点的最大数目：[stepsize]或[stepsize, max_number_vertices]，如果用户没有确定值，MATLAB 使用的默认的值：stepsize = 0.1 (一个单元的十分之一)和最大顶点数目 = 1000。

使用 streamline 命令可以画出由 stream2 返回的数据。

举例：

本例画出了北美地区上空的气流的二维流线图。

```
load wind
[sx,sy] = meshgrid(80,20:10:50);
streamline(stream2(x(:,:,5),y(:,:,5),u(:,:,5),v(:,:,5),sx,sy))
```

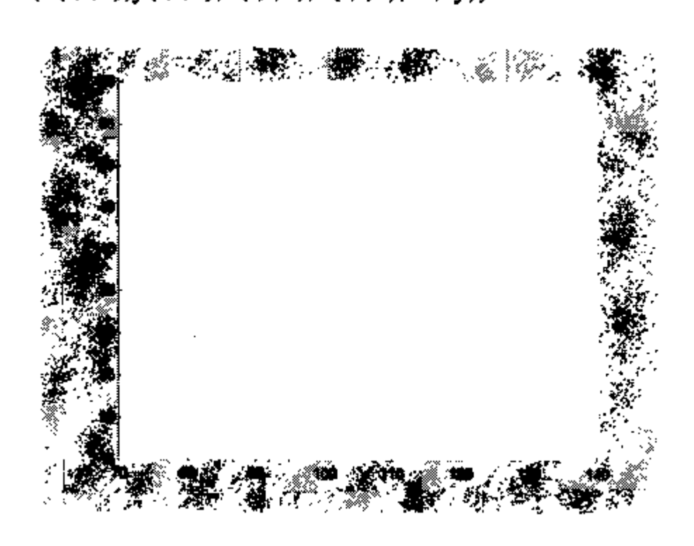


图 16-50　二维流线图

11. 计算三维流线数据

名称：stream3

计算三维流线数据。

语法：该函数有如下两种表达形式：

- XYZ = stream3(X,Y,Z,U,V,W,startx,starty,startz)
- XYZ = stream3(U,V,W,startx,starty,startz)

描述：XYZ = stream3(X,Y,Z,U,V,W,startx,starty,startz)从向量数据 U、V、W 中计算流线。数组 X、Y、Z 定义了 U、V、W 的坐标系，并且必须是单调的和三维网格状的(如由 meshgrid 产生的数据)。startx、starty、和 startz 定义了流线的开始位置。返回值 XYZ 包含顶点数组的一个单元数组。

XYZ = stream3(U,V,W,startx,starty,startz) 假设数组 X、Y 和 Z 被定义为[X,Y,Z] = meshgrid(1:N,1:M,1:P)，这里[M,N,P] = size(U)。

当创建流线时，XYZ = stream3(...,options)确定了使用过的选项。可以定义 options 为一个单元素向量或二元素向量，它包含流线内顶点的阶梯尺寸和顶点的最大数目：[stepsize]或[stepsize, max_number_vertices]如果用户没有确定值，MATLAB 使用的默认的值：stepsize = 0.1（一个单元的十分之一）和最大顶点数目 = 1000。

使用 streamline 命令可以画出由 stream3 返回的数据。

举例：

本例画出了北美地区上空气流的三维流线。

```
load wind
[sx sy sz] = meshgrid(80,20:10:50,0:5:15);
streamline(stream3(x,y,z,u,v,w,sx,sy,sz))
view(3)
```

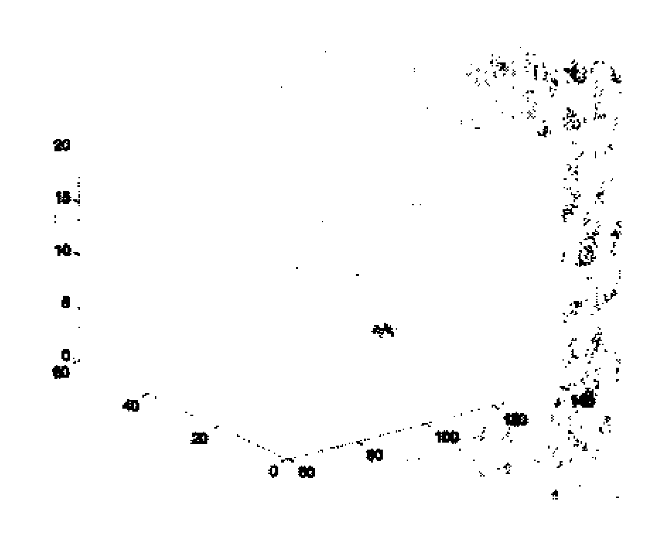


图 16-51　三维流线图

12. 从二维或三维向量数据中画出流线

名称： streamline

从二维或三维向量数据中画出流线。

语法： 该函数有如下七种表达形式：

- h = streamline(X,Y,Z,U,V,W,startx,starty,startz)
- h = streamline(U,V,W,startx,starty,startz)
- h = streamline(XYZ)
- h = streamline(X,Y,U,V,startx,starty)
- h = streamline(U,V,startx,starty)
- h = streamline(XY)
- h = streamline(...,options)

描述： h = streamline(X,Y,Z,U,V,W,startx,starty,startz)从三维向量数据 U、V 和 W 中画出流线图。数组 X、Y、Z 为 U、V、W 定义了坐标系，它们必须是单调和和三维网格状的。startx、

starty、startz 定义了流线的起始位置。输出项 h 包含线句柄的一个向量，每一个流线一个句柄。

h = streamline(U,V,W,startx,starty,startz) 假设了数组 X、Y 和 Z 被定义为[X,Y,Z] = meshgrid(1:N,1:M,1:P)，这里，[M,N,P] = size(U)。

h = streamline(XYZ)假设 XYZ 是一个预先计算过的顶点数组的单元数组。

h = streamline(X,Y,U,V,startx,starty)从二维向量数据 U、V 中画出流线图。数组 X、Y 为 U、V 定义了坐标系，它们必须是单调和和二维网格状的。startx、starty 定义了流线的起始位置。输出项 h 包含线句柄的一个向量，每一个流线一个句柄。

h = streamline(U,V,startx,starty)假设了数组 X、Y 被定义为[X,Y] = meshgrid(1:N,1:M)，这里，[M,N] = size(U)。

h = streamline(XY)假设 XY 是一个预先计算过的顶点数组的单元数组。

streamline(...,options)指定当生成流线时的参数选项，如，一生成条流线的步长或步长和最大顶点数目：[stepsize]或[stepsize, max_number_vertices]，如果用户没有指定这些参数，MATLAB 会使用默认值：stepsize = 0.1 (一个单元的十分之一)和最大顶点数目 = 1000。

举例：

下例绘制了北美某地区上空的气流的流线图。

```
load wind
[sx,sy,sz] = meshgrid(80,20:10:50,0:5:15);
h = streamline(x,y,z,u,v,w,sx,sy,sz);
set(h,'Color','red')
view(3)
```

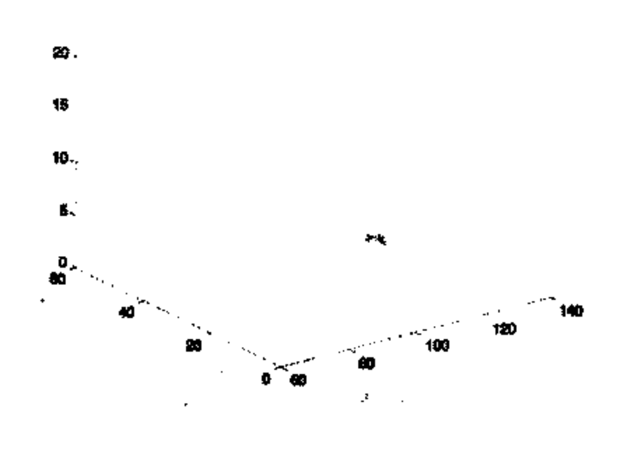


图 16-52　气流的流线图

13. 将表面数据转换为块数据

名称： surf2patch

将表面数据转换为块数据。

语法： 该函数有如下七种表达形式：

- fvc = surf2patch(h)

- fvc = surf2patch(Z)
- fvc = surf2patch(Z,C)
- fvc = surf2patch(X,Y,Z)
- fvc = surf2patch(X,Y,Z,C)
- fvc = surf2patch(...,'triangles')
- [f,v,c] = surf2patch(...)

描述：fvc = surf2patch(h)从句柄 h 确定的表面对象中将几何和颜色数据转化为块格式并返回数据结构 fvc 里的面、顶点和颜色。用户可以直接通过这个数据结构，进入 patch 命令。

fvc = surf2patch(Z)从表面的 ZData 矩阵 Z 中计算块数据。fvc = surf2patch(Z,C)从表面的 ZData 矩阵 Z 和 CData 矩阵 C 中计算块数据。fvc = surf2patch(X,Y,Z)从表面的 XData 矩阵 X、YData 矩阵 Y 和 ZData 矩阵 Z 中计算块数据。fvc = surf2patch(X,Y,Z,C)从表面的 XData、YData、ZData 和 CData 矩阵 X、Y、Z 和 C 中计算块数据。

fvc = surf2patch(...,'triangles')创建组成表面的三角形面，而不是四边形面。

[f,v,c] = surf2patch(...)返回三个数组 f、v 和 c 里的面数据、顶点数据和颜色数据，而不是一个数据结构。

举例：

(1) 第一个例子使用 sphere 命令来生成一个表面的 XData、YData 和 ZData，注意 ZData (z)作为第三和第四选项一起通向 surf2patch，第三个选项是 ZData，第四个选项是 CData。这是因为 patch 命令不能像 surface 命令那样，对颜色数据自动使用 z 坐标的数据。

同样，由于 patch 是一个低层命令，用户必须将视图设置为三维，将阴影设置到面上来产生由 surf 命令产生的同样结果。

```
[x y z] = sphere;
patch(surf2patch(x,y,z,z));
shading faceted; view(3)
```

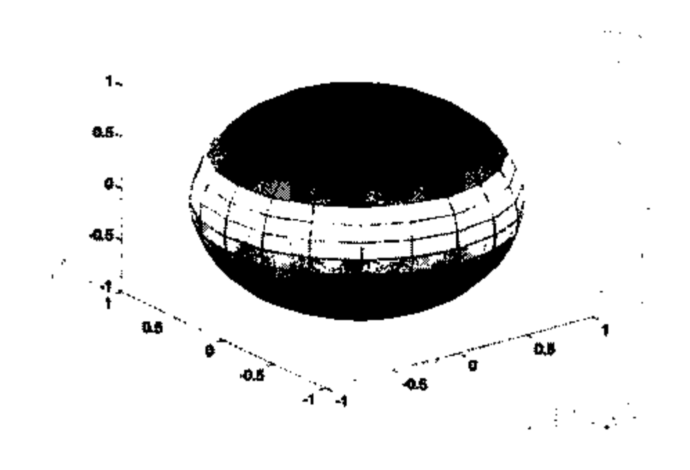


图 16-53　使用 sphere 生成表面图

(2) 第二个例子里，surf2patch 从一个表面计算面，顶点和颜色数据，这个表面的句柄作为一个选项已经通过了。

```
s = surf(peaks);
pause
patch(surf2patch(s));
delete(s)
shading faceted; view(3)
```

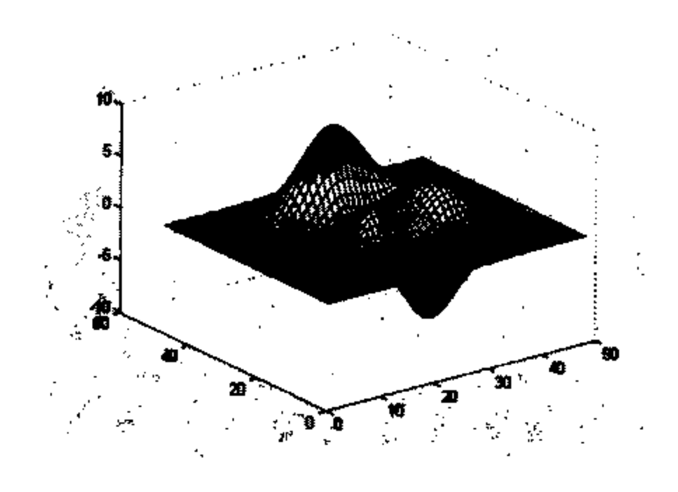


图 16-54　peaks 生成的表面图

14. 提取子集

名称：subvolume

从体数据集合中提取子集。

语法：该函数有如下三种表达形式：

- [Nx,Ny,Nz,Nv] = subvolume(X,Y,Z,V,limits)
- [Nx,Ny,Nz,Nv] = subvolume(V,limits)
- Nv = subvolume(...)

描述：[Nx,Ny,Nz,Nv] = subvolume(X,Y,Z,V,limits)使用指定的轴的范围从体数据集里提取一个子集，limits = [xmin,xmax,ymin, ymax,zmin,zmax]定义了坐标轴的区间。

数组 X、Y 和 Z 为体 V 定义了坐标系。subvolume 被返回在 NV 里，子体的坐标系在 NX、NY 和 NZ 里给出。

[Nx,Ny,Nz,Nv] = subvolume(V,limits)假设了数组 X、Y 和 Z 被定义为[X,Y,Z] = meshgrid(1:N,1:M,1:P)，这里[M,N,P] = size(V)。

Nv = subvolume(...)只返回子体。

举例：

把一个四维数组压缩为三维，然后用 subvolume 函数从中提取出其子集。

```
load mri
D = squeeze(D);
[x,y,z,D] = subvolume(D,[60,80,nan,80,nan,nan]);
p1 = patch(isosurface(x,y,z,D, 5),...
     'FaceColor','red','EdgeColor','none');
```

```
isonormals(x,y,z,D,p1);
p2 = patch(isocaps(x,y,z,D, 5),...
       'FaceColor','interp','EdgeColor','none');
view(3); axis tight; daspect([1,1,.4])
colormap(gray(100))
camlight right; camlight left; lighting gouraud
```

图 16-55　人头骨图

16.6　域生成

1. 数据网格化

名称：griddata

数据网格化。

语法：该函数有如下三种表达形式：

- ZI = griddata(x,y,z,XI,YI)
- [XI,YI,ZI] = griddata(x,y,z,xi,yi)
- [...] = griddata(...,method)

描述：ZI = griddata(x,y,z,XI,YI)以形式为 z = f(x,y)的表面拟合非等间距向量里的数据。griddata 在由(XI,YI)确定的点处插值这个表面来产生 ZI。这个表面通过数据点。XI 和 YI 经常形成一个均匀的网格(正如 meshgrid 产生的那样)。

XI 可以是一个行向量，在这种情况下它确定了一个具有常数列的矩阵。相似地，YI 可以是一个列向量，它确定了一个具有常数行的矩阵。

[XI,YI,ZI] = griddata(x,y,z,xi,yi)返回插值矩阵 ZI，也返回从行向量 xi 和列向量 yi 形成的矩阵 XI 和 YI。后者和由 meshgrid 返回的矩阵是相同的。

[...] = griddata(...,method)使用指定的插值方法：'linear'，'cubic'，'nearest'和'v4'。插值方法定义了拟合数据的表面的类型。当'linear'和'nearest'分别有不连续的一阶和零阶导数时，

'cubic'和'v4'方法可以产生光滑的表面。处理'v4'外的所有方法都是基于数据的 Delaunay 三角化的

举例：

在±2.0 之间的 100 个点出取样一个函数：

```
rand('seed',0)
x = rand(100,1)*4-2; y = rand(100,1)*4-2;
z = x.*exp(-x.^2-y.^2);
```

x、y 和 z 现在是包含非均匀的样本数据。定义一个矩形网格并将赋予数据：

```
ti = -2:.25:2;
[XI,YI] = meshgrid(ti,ti);
ZI = griddata(x,y,z,XI,YI);
```

沿非均匀数据点画成网格数据：

```
mesh(XI,YI,ZI), hold
plot3(x,y,z,'o'), hold off
```



图 16-56　数据网格化

2. 为三维图形生成 X 和 Y 矩阵

名称：meshgrid

为三维图形生成 X 和 Y 矩阵。

语法：该函数有如下三种表达形式：

- [X,Y] = meshgrid(x,y)
- [X,Y] = meshgrid(x)
- [X,Y,Z] = meshgrid(x,y,z)

描述：[X,Y] = meshgrid(x,y)变换向量 x 和 y 定义的区域得到数组 X 和 Y，可用来绘制三维 mesh/surface 图形。数组 X 的行是向量 x 的拷贝，数组 Y 的列是向量 y 的拷贝。

[X,Y] = meshgrid(x)等价于[X,Y] = meshgrid(x,x)。

[X,Y,Z] = meshgrid(x,y,z)生成三维数组，可用来对三变量的函数估值或绘制三维立体图。

举例：

```
[X,Y] = meshgrid(1:3,10:14)
X =
     1     2     3
     1     2     3
     1     2     3
     1     2     3
     1     2     3
Y =
    10    10    10
    11    11    11
    12    12    12
    13    13    13
    14    14    14
```

16.7 专门图形绘制

1. 一个二维图形的区域填充

名称：area

一个二维图形的区域填充。

语法：该函数有如下五种表达形式：

- area(Y)
- area(X,Y)
- area(...,ymin)
- area(...,'PropertyName',PropertyValue,...)
- h = area(...)

描述：绘图命令 area 将 Y 中的元素显示为一个或多个曲线并填充每一条曲线下的区域。当 Y 是矩阵时，曲线被堆叠起来，显示在每个 x 间隔内，每一个行元素对曲线的总高度的相对贡献。

area(Y)画出向量 Y 或矩阵 Y 里每一列的和。当 Y 是向量时，x 轴根据 length(Y)进行自动调整，当 Y 是矩阵时，x 轴根据 size(Y,1)进行自动调整。

area(X,Y)在 X 的对应的值处绘出 Y。如果 X 是一个向量，length(X)必须等于 length(Y)并且 X 必须是单调的。如果 X 是一个矩阵，size(X)必须等于 size(Y)并且 X 的每一列必须是单调的。sort 命令可以使一个向量或矩阵单调。

area(...,ymin)为区域填充确定 y 方向的下限。默认的 ymin 是 0。

area(...,'PropertyName',PropertyValue,...)为由 area 创建的块图形对象确定属性名和属性值这一选项对。

h = area(...)返回一个指向块图形对象的句柄。area 在 Y 中每一列创建一个块对象。

举例：

将 Y 中的值画成一个堆叠的区域图。

```
Y = [    1, 2, 4;
         3, 5, 7;
         1, 7, 3;
         2, 5, 1];
area(Y)
grid on
colormap summer
set(gca,'Layer','top')
title 'Stacked Area Plot'
```

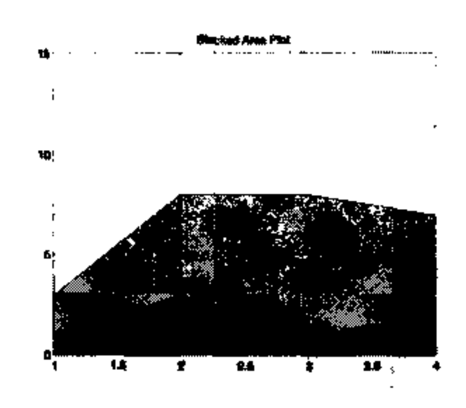


图 16-57　堆叠的 area 图

2. 控制轴的边界

名称：box

控制轴的边界。

语法：该函数有如下三种表达形式：

- box on
- box off
- box

描述：box on 显示当前坐标轴的边界；box off 不显示当前坐标轴的边界；box 切换当前坐标轴的显示状态。

3. 二维彗星轨迹图

名称：comet

二维彗星轨迹图。

语法：该函数有如下三种表达形式：

- comet(y)
- comet(x,y)
- comet(x,y,p)

描述：一个彗星状轨迹绘图命令是一个动画，其中一个圆(彗星头)会在屏幕上追踪数据点。彗星体是跟随彗星头的轨迹段。彗星尾是一条追踪整个函数的实心线。

comet(y)显示向量 y 的彗星状轨迹图。comet(x,y)显示向量 y 相对于向量 x 的彗星状轨迹图。comet(x,y,p)确定一个长度为 p*length(y)的彗星体。p 默认值是 0.1。

举例:

创建一个简单的彗星轨迹图。

```
t = 0:.01:2*pi;
x = cos(2*t).*(cos(t).^2);
y = sin(2*t).*(sin(t).^2);
comet(x,y);
```

4. 绘制原点出发的向量图

名称：compass

绘制原点出发的向量图。

语法：该函数有如下四种表达形式：

- compass(X,Y)
- compass(Z)
- compass(...,LineSpec)
- h = compass(...)

描述：compass 绘制从原点向外发出的向量图。X，Y 和 Z 都是笛卡尔直角坐标，图形绘制在圆周网格图上。

compass(X,Y)绘制有 n 个箭头的向量图，其中 n 是 X 或 Y 中元素的个数。每个箭头都由原点发出。箭头端点所在点的坐标是[X(i),Y(i)]。

compass(Z)绘制有 n 个箭头的复数向量图，其中 n 是 Z 中元素的个数。每个箭头都由原点发出。箭头端点所在点的坐标由 Z 的实部和虚部确定，即等价于 compass(real(Z), imag(Z))。

compass(...,LineSpec)绘制的向量图的线型，标记符号和颜色由 LineSpec 定义。

h = compass(...)返回指向线对象的句柄。

举例:

绘制一个矩阵特征向量的向量图。

```
Z = eig(randn(10,10));
compass(Z)
```

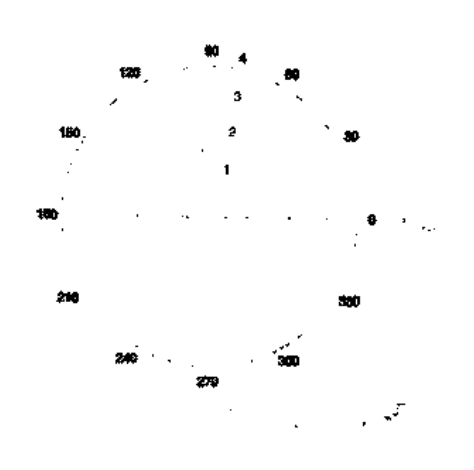


图 16-58 一个矩阵特征向量的向量图

5. 简易等高线图绘图仪

名称：ezcontour

简易等高线图绘图仪。

语法：该函数有如下三种表达形式：

- ezcontour(f)
- ezcontour(f,domain)
- ezcontour(...,n)

描述：ezcontour(f)绘制 f(x,y)的等高线图，其中 f 是一个以字符串形式给出的两变量如 x 和 y 的数学函数。

f 绘制在默认的$-2\pi < x < 2\pi$，$-2\pi < y < 2\pi$区域上。MATLAB 根据变化范围的大小选择计算的网格数；如果函数 f 在某些网格点上没有定义，则不绘制这些点。

ezcontour(f,domain)在指定区域上绘制 f(x,y)的等高线图。该区域可以是由 4×1 的向量[xmin, xmax, ymin, ymax]确定或由 2×1 的向量[min, max]确定。(即 min < x < max, min < y < max)

如果函数 f 的自变量不是 x 和 y，而是 u 和 v，则自变量定义域的端点 umin, umax, vmin, and vmax 按字母顺序排列取值。这样，ezcontour('u^2 - v^3',[0,1],[3,6])在 $0 < u < 1$，$3 < v < 6$ 区域上绘制等高线图。

ezcontour(...,n)在默认区域上绘图，其坐标网格数为 n×n。n 的默认值是 60。

ezcontour 自动在图上添加标题和轴标。

举例：

下面的表达式定义一个两自变量 x 和 y 的函数：

$$f(x,y) = 3(1-x)^2*\exp(-(x^2)-(y+1)^2)-10(x/3-x^3-y^5)*\exp(-x^2-y^2)-1/3*\exp(-x^2-y^2)$$

ezcontour 要求用一个字符串来表达该函数，且要按照 MATLAB 的语法描述指数函数，自然对数等函数的方法来表达：

```
f = ['3*(1-x)^2*exp(-(x^2)-(y+1)^2)',...
     '- 10*(x/5 - x^3 - y^5)*exp(-x^2-y^2)',...
     '- 1/3*exp(-(x+1)^2 - y^2)'];
ezcontour(f,[-3,3],49)
```

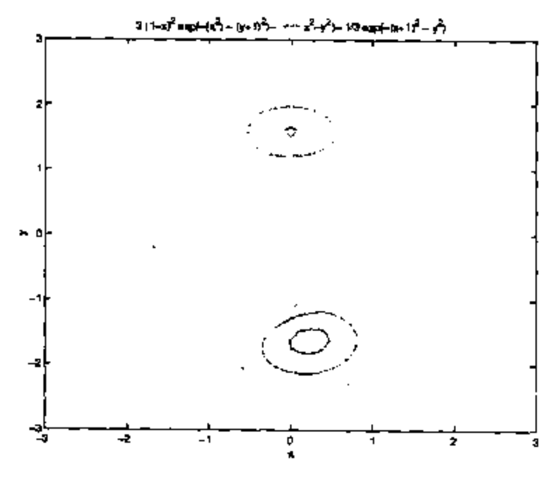


图 16-59　给定函数的等高线图

6. 简易填充等高线绘图仪

名称：ezcontourf

简易填充等高线绘图仪。

语法：该函数有如下三种表达形式：

- ezcontourf(f)
- ezcontourf(f,domain)
- ezcontourf(...,n)

描述：ezcontourf(f)画出 f(x,y)的等高线，这里 f 是表示一个含双变量(如 x 和 y)的数学函数的字符串。

函数 f 的默认定义域为$-2\pi < x < 2\pi$，$-2\pi < y < 2\pi$。MATLAB 根据变化的幅度选择计算网格，如果函数 f 没有定义网格上的点，则这些点就不画出。

ezcontourf(f,domain)在指定的定义域上画出 f(x,y)，定义域或者是 4×1 向量[xmin, xmax, ymin, ymax]，或者是 2×1 向量[min,max](这里，min < x < max, min < y < max)。

如果 f 是变量 u 和 v(而不是 x 和 y)的函数，则定义域的端点为 umin、umax、vmin 和 vmax，它们是按字母顺序排列的。因此，ezcontourf('u^2 - v^3',[0,1],[3,6])画出的是 $u^2 - v^3$ 在 $0 < u < 1$，$3 < v < 6$ 上的等高线。

ezcontourf(...,n)画出在默认使用 n×n 网格的定义域上的 f。n 的默认值是 60。

ezcontourf 自动增加一个标题和轴标。

举例：

下面的数学表达式定义了含变量 x 和 y 的一个函数：

$f(x,y) = 3(1-x)^2*\exp(-(x^2)-(y+1)^2)-10(x/3-x^3-y^5)*\exp(-x^2-y^2)-1/3*\exp(-x^2-y^2)$

这个函数可以通过字符串表示为：

```
f = ['3*(1-x)^2*exp(-(x^2)-(y+1)^2)',...
     '- 10*(x/3 - x^3 - y^5)*exp(-x^2-y^2)',...
     '- 1/3*exp(-(x)^2 - y^2)'];
```

将字符串变量 f 沿-3 到 3 的范围送往 ezcontourf，并指定 49×49 的网格：

```
ezcontourf(f,[-3,3],49)
```

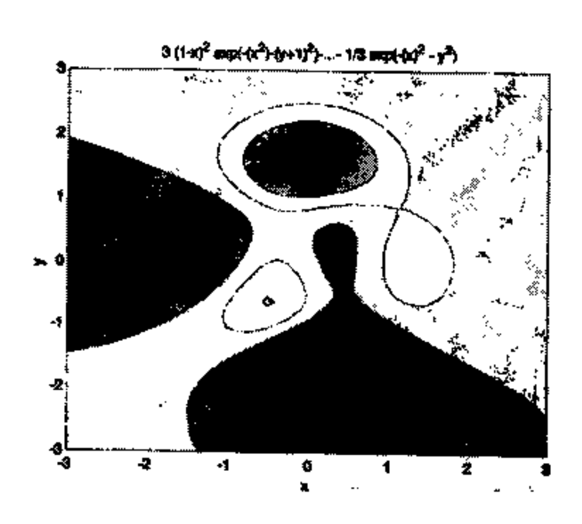


图 16-60　给定函数的等高线图

7. 简易网线图绘图仪

名称：ezmesh

简易网线图绘图仪。

语法：该函数有如下六种表达形式：

- ezmesh(f)
- ezmesh(f,domain)
- ezmesh(x,y,z)
- ezmesh(x,y,z,[smin,smax,tmin,tmax]) or ezmesh(x,y,z,[min,max])
- ezmesh(...,n)
- ezmesh(...,'circ')

描述：ezmesh(f)绘制 f(x,y)的网线图，其中 f 是一个以字符串形式给出的两变量如 x 和 y 的数学函数。

函数 f 绘制在默认的$-2\pi < x < 2\pi$，$-2\pi < y < 2\pi$区域上。MATLAB 根据变化范围的大小选择计算的网格数；如果函数 f 在某些网格点上没有定义，则不绘制这些点。

ezmesh(f,domain)在指定区域上绘制 f(x,y)的网线图。该区域可以是由 4×1 的向量 [xmin, xmax, ymin, ymax]确定或由 2×1 的向量[min, max]确定。(即 min < x < max, min < y < max)

如果函数 f 的自变量不是 x 和 y，而是 u 和 v，则自变量定义域的端点 umin, umax, vmin, and vmax 按字母顺序排列取值。这样，ezmesh('u^2 - v^3',[0,1],[3,6])在 $0 < u < 1$，$3 < v < 6$ 区域上绘制网线图。

ezmesh(x,y,z)在正方形区域$-2\pi < s < 2\pi$，$-2\pi < t < 2\pi$.上绘制参数曲面 x = x(s,t), y = y(s,t), 和 z = z(s,t)的网线图。

ezmesh(x,y,z,[smin,smax,tmin,tmax])或 ezmesh(x,y,z,[min,max])在指定区域是绘制参数曲面的网线图。

ezmesh(...,n)在默认区域是绘制 f 的网线图，其坐标网格数为 n×n。n 的默认值是 60。

ezmesh(...,'circ')在区域的中心圆盘上绘制 f 的网线图。

举例：

下例可视化函数：f(x, y) = x*exp(-x*x – y*y)。

```
ezmesh('x*exp(-x^2-y^2)',40)
colormap [0 0 1]
```

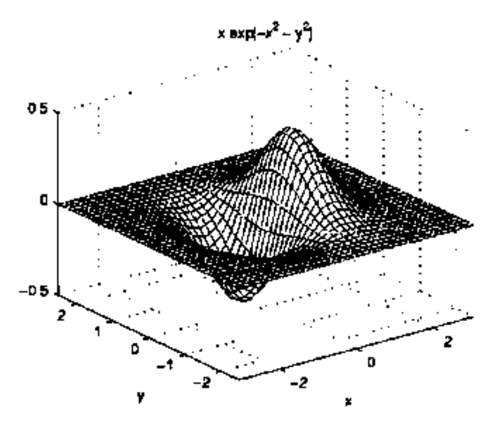


图 16-61 函数可视化图

8. 简易网线/等高线组合绘图仪

名称：ezmeshc

简易网线/等高线组合绘图仪。

语法：该函数有如下六种表达形式：

- ezmeshc(f)
- ezmeshc(f,domain)
- ezmeshc(x,y,z)
- ezmeshc(x,y,z,[smin,smax,tmin,tmax]) or ezmeshc(x,y,z,[min,max])
- ezmeshc(...,n)
- ezmeshc(...,'circ')

描述：ezmeshc(f)创建 f(x,y)的图形，这里 f 是表示含两个变量，如 x 和 y，的数学函数的一个串。

函数 f 的默认定义域为$-2\pi < x < 2\pi$，$-2\pi < y < 2\pi$。MATLAB 根据变化的幅度选择计算网格，如果函数 f 没有定义网格上的点，则这些点就不画出。

ezcontourf(f,domain)在指定的定义域上画出 f(x,y)，定义域或者是 4×1 向量[xmin, xmax, ymin, ymax]，或者是 2×1 向量[min,max](这里，min < x < max, min < y < max)。

如果 f 是变量 u 和 v(而不是 x 和 y)的函数，则定义域的端点为 umin、umax、vmin 和 vmax，它们是按字母顺序排列的。因此，ezcontourf('u^2 - v^3',[0,1],[3,6])画出的是 $u^2 - v^3$ 在 $0 < u < 1$，$3 < v < 6$ 上的等高线。

ezmeshc(x,y,z)在矩形域$-2\pi < s < 2\pi$，$-2\pi < t < 2\pi$上画出参数表面 x = x(s,t)、y = y(s,t) 和 z = z(s,t)。

ezmeshc(x,y,z,[smin,smax,tmin,tmax])或 ezmeshc(x,y,z,[min,max])在指定的定义域上画出参数表面。

ezmeshc(...,n)在默认的使用 n×n 网格的定义域上画出 f。n 的默认值是 60。

ezmeshc(...,'circ')在以定义域内一点为中心的圆盘上画出 f。

举例：

在定义域-5 < x < 5, -2*pi < y < 2*pi 上创建函数 $f(x,y) = y / (1 + x^2 + y^2)$的一个网线/等高线图。

```
ezmeshc('y/(1 + x^2 + y^2)',[-5,5,-2*pi,2*pi])
```

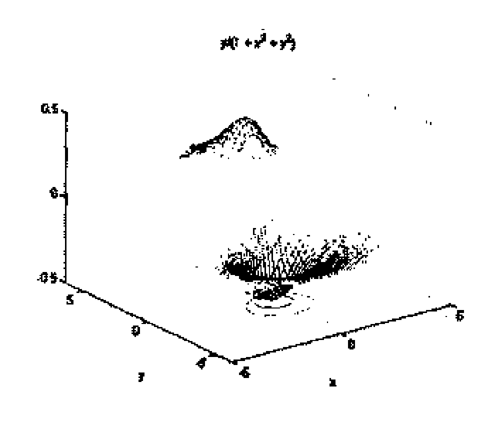


图 16-62　函数的网线/等高线图

9. 简易曲线图绘图仪

名称：ezplot

简易曲线图绘图仪。

语法：该函数有如下六种表达形式：

- ezplot(f)
- ezplot(f,[min,max])
- ezplot(f,[xmin,xmax,ymin,ymax])
- ezplot(x,y)
- ezplot(x,y,[tmin,tmax])
- ezplot(...,figure)

描述：ezplot(f)在默认区域$-2\pi < x < 2\pi$是绘制函数 $f = f(x)$的曲线图。

ezplot(f,[min,max])在区域 min < x < max 是绘制函数 $f = f(x)$的曲线图。

对于隐式函数 $f = f(x,y)$：ezplot(f)在$-2\pi < x < 2\pi$，$-2\pi < y < 2\pi$区域上绘制 $f(x,y) = 0$ 的曲线图。

ezplot(f,[xmin,xmax,ymin,ymax]) 在 xmin < x < xmax 和 ymin < y < ymax 上绘制 $f(x,y) = 0$ 曲线图。

ezplot(f,[min,max])在 min < x < max 和 min < y < max 区域上绘制 $f(x,y) = 0$ 的曲线图。

如果函数 f 的自变量不是 x 和 y，而是 u 和 v，则自变量定义域的端点 umin, umax, vmin, and vmax 按字母顺序排列取值。这样，ezplot('u^2 - v^2 - 1',[-3,2,-2,3])在$-3 < u < 2$，$-2 < v < 3$区域上绘制曲线图。

ezplot(x,y)在默认区域 $0 < t < 2\pi$上绘制参数曲线 $x = x(t)$和 $y = y(t)$的曲线图。

ezplot(x,y,[tmin,tmax])在区域 tmin < t < tmax 上绘制参数曲线 $x = x(t)$和 $y = y(t)$的曲线图。

ezplot(...,figure)在给定句柄所指向的图形窗口中绘制曲线图。

举例：

绘制隐式函数：

$x^2 - y^4 = 0$

在[-2, 2]区域上：

```
ezplot('x^2-y^4')
```

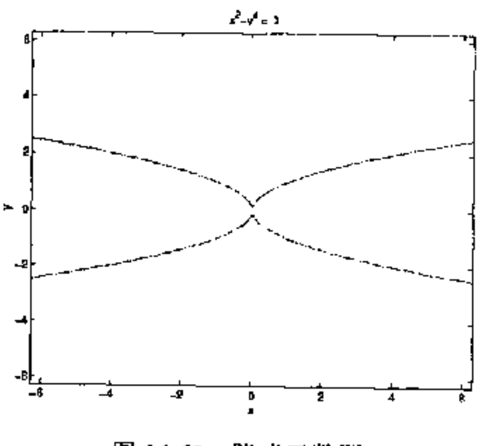


图 16-63　隐式函数图

10. 简易三维参数曲线绘图仪

名称：ezplot3

简易三维参数曲线绘图仪。

语法：该函数有如三下种表达形式：

- ezplot3(x,y,z)
- ezplot3(x,y,z,[tmin,tmax])
- ezplot3(...,'animate')

描述：ezplot3(x,y,z)在默认定义域 $0 < t < 2\pi$上画出空间曲线 x = x(t)、y = y(t)和 z = z(t)。ezplot3(x,y,z,[tmin,tmax])在定义域 tmin < t < tmax 上画出曲线 x = x(t)、y = y(t)和 z = z(t)。ezplot3(...,'animate')产生空间曲线的一个动画轨迹。

举例：

本例在定义域[0,10]上画出一个参数曲线。

ezplot3('cos(t)','sin(t)','t^2',[0,10*pi])

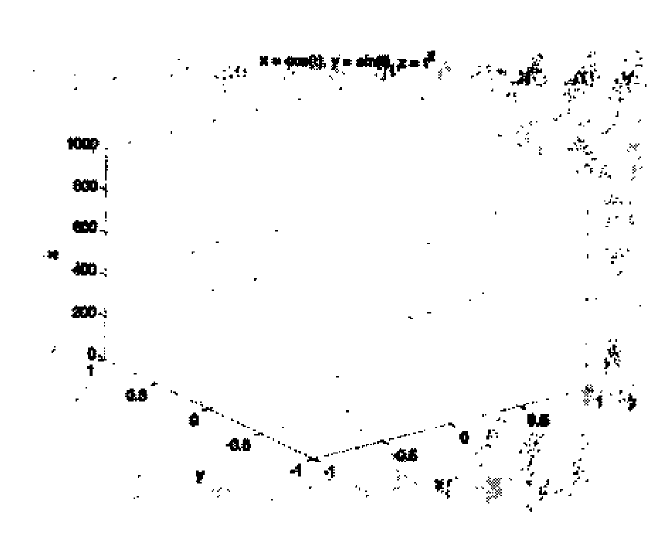


图 16-64 绘制参数曲线图

11. 简易三维着色表面绘图仪

名称：ezsurf

简易三维着色表面绘图仪。

语法：该函数有如下六种表达形式：

- ezsurf(f)
- ezsurf(f,domain)
- ezsurf(x,y,z)
- ezsurf(x,y,z,[smin,smax,tmin,tmax]) or ezsurf(x,y,z,[min,max])
- ezsurf(...,n)
- ezsurf(...,'circ')

描述：ezsurf(f)创建 f(x,y)的图形，这里 f 是两个变量 x 和 y 的数学函数的字符串。

函数 f 的默认定义域是$-2\pi < x < 2\pi$，$-2\pi < y < 2\pi$。MATLAB 根据变化的幅度选择计算网格，如果函数 f 没有定义网格上的点，则这些点就不画出。

ezsurf(f,domain)在指定的定义域上画出 f(x,y)，定义域或者是 4×1 向量[xmin, xmax, ymin, ymax]，或者是 2×1 向量[min,max](这里，min < x < max, min < y < max)。

如果 f 是变量 u 和 v(而不是 x 和 y)的函数，则定义域的端点为 umin, umax, vmin, and vmax，它们是按字母顺序排列的。因此，ezcontourf('u^2 - v^3',[0,1],[3,6])画出的是 $u^2 - v^3$ 在 $0 < u < 1$，$3 < v < 6$ 上的等高线。

ezsurf(x,y,z)画出在默认定义域 $0 < t < 2\pi$上的空间曲线 x = x(t)、y = y(t)和 z = z(t)。ezsurf(x,y,z,[smin,smax,tmin,tmax])或 ezsurf(x,y,z,[min,max])使用指定定义域画出参数表面。

ezsurf(...,n)在默认的使用 n×n 网格的定义域上画出 f。n 的默认值是 60。

ezsurf(...,'circ') 在以定义域内一点为中心的圆盘上画出 f。

举例：

(1) 画出在默认定义域$-2\pi < x < 2\pi$，$-2\pi < y < 2\pi$的函数：f(x,y) = real(atan(x + i * y))。

```
ezsurf('real(atan(x+i*y))')
```

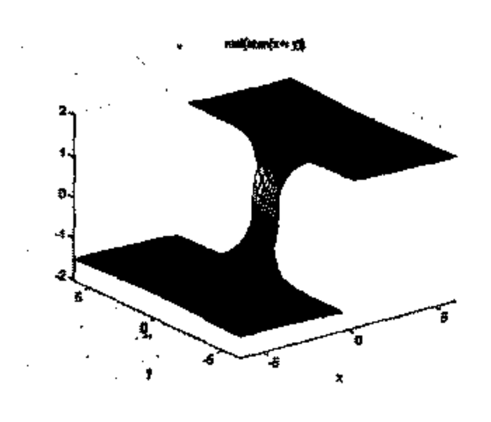


图 16-65　函数的三维着色图

(2) 不开启非连续的过滤作用，使用 surf 根据同样的数据画出图形。

```
[x,y] = meshgrid(linspace(-2*pi,2*pi,60));
z = real(atan(x+i.*y));
surf(x,y,z)
```

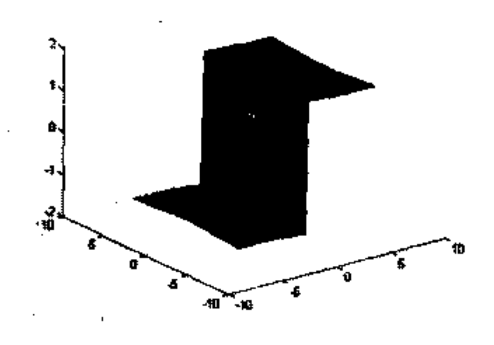


图 16-66　函数的三维着色图

12. 简易极坐标曲线图

名称：ezpolar

简易极坐标曲线图。

语法：该函数有如下两种表达形式：

- ezpolar(f)
- ezpolar(f,[a,b])

描述：ezpolar(f)在默认区域 0 < theta < 2π.上绘制极坐标曲线 rho = f(theta)。ezpolar(f,[a,b])在区域 a < theta < b 上绘制极坐标曲线 rho = f(theta)。

举例：

绘制一个极坐标曲线图。

1 + sin(t)

在区域[0, 1]上绘制：

ezpolar('1+sin(t)')

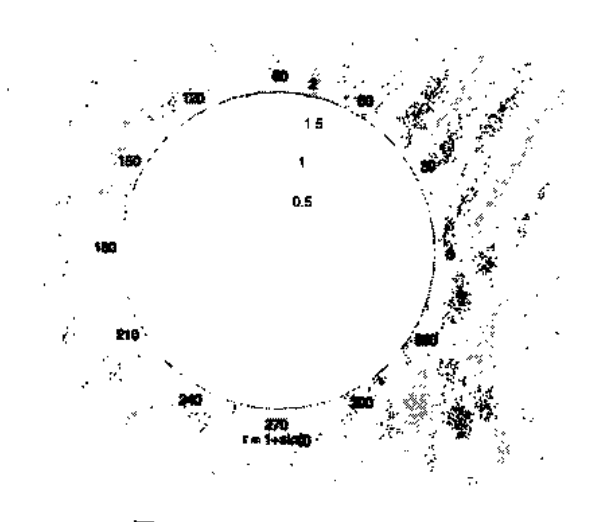


图 16-67　极坐标曲线图

13. 沿水平轴等间距的点发散的向量

名称：feather

沿水平轴等间距的点发散的向量。

语法：该函数有如下三种表达形式：

- feather(U,V)
- feather(Z)
- feather(...,LineSpec)

描述：一个 feather 命令显示的是沿水平轴等间距的点发散的向量。用户可以表示相对于各自向量的初始点的向量分量。

feather(U,V)显示由 U 和 V 指定的向量，这里 U 包含 x 分量作为相关的坐标系，V 包含 y 分量作为相关的坐标系。

feather(Z)显示由 Z 中的复数确定的向量。它等效于 feather(real(Z),imag(Z))。

feather(...,LineSpec)使用由 LineSpec 确定的线型，标记符号和颜色。

举例：

创建一个显示 theta 方向的羽毛状图形。

```
theta = (-90:10:90)*pi/180;
r = 4*ones(size(theta));
[u,v] = pol2cart(theta,r);
feather(u,v);
```

4
3
2
1
0
-1
-2
-3
-4
0 2 4 6 8 10 12 14 16 18 20

图 16-68　羽毛状图

14. 简易带等高线的三维表面图绘图仪

名称：ezsurfc

简易带等高线的三维表面图绘图仪。

语法：该函数有如下六种表达形式：

- ezsurfc(f)
- ezsurfc(f,domain)
- ezsurfc(x,y,z)
- ezsurfc(x,y,z,[smin,smax,tmin,tmax]) or ezsurfc(x,y,z,[min,max])
- ezsurfc(...,n)
- ezsurfc(...,'circ')

描述：ezsurfc(f) 绘制 f(x,y)的带等高线的三维表面图，其中 f 是一个以字符串形式给出的两变量如 x 和 y 的数学函数。

函数 f 绘制在默认的$-2\pi < x < 2\pi$，$-2\pi < y < 2\pi$区域上。MATLAB 根据变化范围的大小选择计算的网格数；如果函数 f 在某些网格点上没有定义，则不绘制这些点。

ezsurfc(f,domain)在指定区域上绘制 f(x,y)的带等高线的三维表面图。该区域可以是由 4×1 的向量 [xmin, xmax, ymin, ymax]确定或由 2×1 的向量[min, max]确定。(即 min < x < max, min < y < max)

如果函数 f 的自变量不是 x 和 y，而是 u 和 v，则自变量定义域的端点 umin, umax, vmin, and vmax 按字母顺序排列取值。这样，ezsurfc ('u^2 - v^3',[0,1],[3,6])在 $0 < u < 1$，$3 < v < 6$ 区域上绘制带等高线的三维表面图。

ezsurfc(x,y,z) 在正方形区域$-2\pi < s < 2\pi$，$-2\pi < t < 2\pi$上绘制参数曲面 x = x(s,t), y = y(s,t), 和 z = z(s,t)的带等高线的三维表面图。

ezsurfc(x,y,z,[smin,smax,tmin,tmax])或 ezsurfc(x,y,z,[min,max])在指定区域上绘制参数曲面的带等高线的三维表面图。

ezsurfc(...,n) 在默认区域是绘制 f 的带等高线的三维表面图，其坐标网格数为 n×n。n 的默认值是 60。

ezsurfc(...,'circ')在区域的中心圆盘上绘制 f 的带等高线的三维表面图。

举例：

在区域-5 < x < 5, -2*pi < y < 2*pi 上绘制一个表达式为 f(x, y) = y / (1 + x*x + y*y) 的带等高线的三维表面图。

ezsurfc('y/(1 + x^2 + y^2)',[-5,5,-2*pi,2*pi],35)

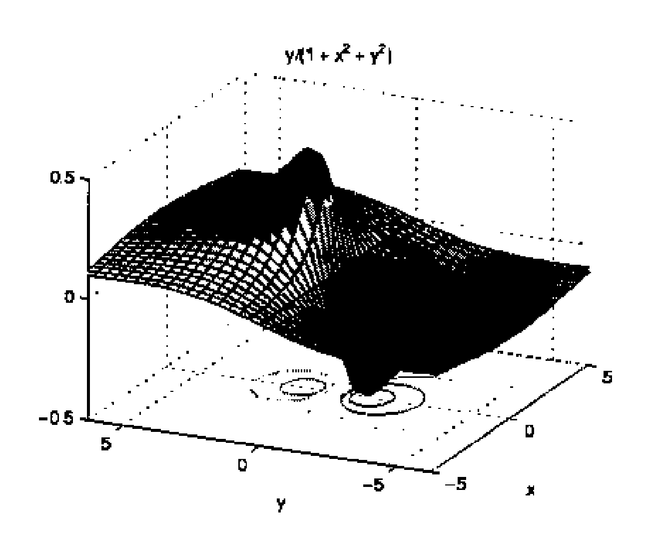


图 16-69　带等高线的三维表面图

15. 在指定的区间内画出一个函数的图形

名称： fplot

在指定的区间内画出一个函数的图形。

语法： 该函数有如下五种表达形式：

- fplot('function',limits)
- fplot('function',limits,LineSpec)
- fplot('function',limits,tol)
- fplot('function',limits,tol,LineSpec)
- [x,Y] = fplot(...)

描述： fplot 在指定的区间画出一个函数。这个函数必须形如 y = f(x)，这里 x 是一个指定了区间的向量，y 是与 x 有相同维数的一个向量，并且是 x 中的某点的函数值。如果这个函数对给定的 x 返回多于一个的值，则 y 是一个矩阵，这个矩阵的列包含 f(x)的每一个分量。

fplot('function',limits)在由 limits 确定的区间内画出'function'。limits 是一个在 x 轴上确定了界限([xmin xmax])的向量，或在 x 轴和 y 轴上的确定了界限([xmin xmax ymin ymax])的函数。

fplot('function',limits,LineSpec)使用 LineSpec 确定的线画出'function'。'function'是一个 MATLAB 的 M 文件的名字或包含变量 x 的一个字符串。

fplot('function',limits,tol)使用相对允许误差 tol 画出'function'(默认是 2e-3)，x 的最大步数是(1/tol)+1。

fplot('function',limits,tol,LineSpec)使用相对允许误差 tol 和一个指定了线型、标记和颜色的线型规范画出'function'。

[x,Y] = tplot(...)为 x 和 Y 中的'function'返回横坐标和纵坐标。在屏幕上并不画出图形。用户使用 plot(x,Y)画出这个函数。

举例：

画出-2 到 2 的双曲正弦函数。

fplot('sinh',[-2 2])

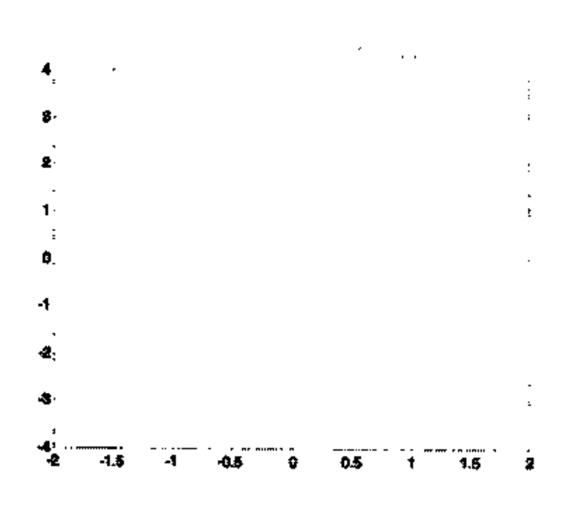


图 16-70　双曲正弦函数图

16. 填充二维多边形

名称： fill

填充二维多边形。

语法： 该函数有如下五种表达形式：

- fill(X,Y,C)
- fill(X,Y,ColorSpec)
- fill(X1,Y1,C1,X2,Y2,C2,...)
- fill(...,'PropertyName',PropertyValue)
- h = fill(...)

描述： fill 函数生成着色的多边形。

fill(X,Y,C)生成由 X 和 Y 中的数据所得的多边形，并根据 C 所指定的顶点颜色进行填充。C 是一个向量或矩阵，被用作色图的索引。如果 C 是行向量，length(C)必须等于 size(X,2)和 size(Y,2)；如果 C 是列向量，length(C)必须等于 size(X,1)和 size(Y,1)。如果有必要，fill 函数能连接最后顶点和初始的顶点以封闭多边形。

fill(X,Y,ColorSpec)填充由 X 和 Y 定义的二维多边形，其填充色由 ColorSpec 指定。

fill(X1,Y1,C1,X2,Y2,C2,...)填充多个二维多边形。

fill(...,'PropertyName',PropertyValue)允许用户指定一个块图形对象的属性名称和数值。

h = fill(...)返回一个指向块图形对象的句柄向量。

举例：

绘制一个红色八边形：

```
t = (1/16:1/8:1)'*2*pi;
x = sin(t);
y = cos(t);
fill(x,y,'r')
axis square
```

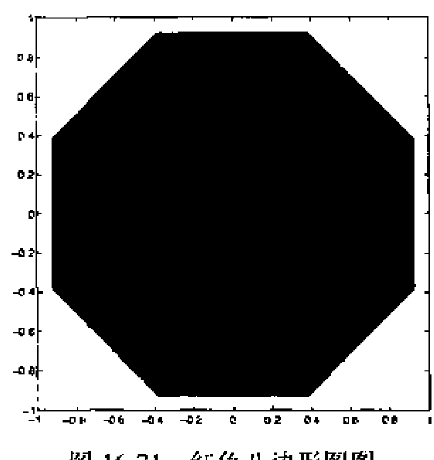


图 16-71　红色八边形图图

17. 三维饼图

名称：pie3

三维饼图。

语法：该函数有如下三种表达形式：

- pie3(X)
- pie3(X,explode)
- h = pie3(...)

描述：pie3(X)使用 X 里的数据画出一个三维饼图。X 里的每一个元素在饼图里用切片来表示。

pie3(X.explode)确定是否从饼图的中心分出一个切片。如果 explode(i,j)不等于零，X(i,j)就是从饼图的中心分出的分支。explode 必须与 X 有相同的阶数。

h = pie(...)返回一个指向块、表面、和文本图形对象的句柄向量。

举例：

通过将对应的 explode 元素设置为 1 来画出一个从饼图里分离的切片。

```
x = [1 5 4 3 2]
explode = [0 1 0 0 0]
pie3(x,explode)
colormap hsv
```

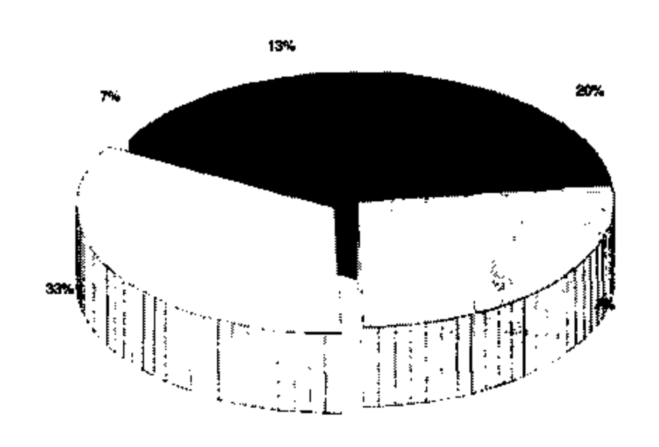


图 16-72 三维饼图

18. Pareto 图

名称：pareto

Pareto 图。

语法：该函数有如四下种表达形式：

- pareto(Y)
- pareto(Y,names)
- pareto(Y,X)
- H = pareto(...)

描述：Pareto charts 以直条的形式递减地显示向量 Y 中的数值。

pareto(Y)在直条上用 Y 的元素的下标进行标注。

pareto(Y,names)用字符串矩阵中相应的名称标记每个直条。

pareto(Y,X)从 X 中的相关取值对直条进行标记。

H = pareto(...)返回一个指向块对象和线对象的联合的句柄。

19. 绘制离散图

名称：plotmatrix

绘制离散图。

语法：该函数有如下三种表达形式

- plotmatrix(X,Y)
- plotmatrix(...,'LineSpec')
- [H,AX,BigAx,P] = plotmatrix(...)

描述：plotmatrix(X,Y)绘制 X 的列对 Y 的列的离散图。如果 X 是 p×n 的矩阵，Y 是 p×n 的矩阵，则 plotmatrix 绘制 n×m 个子图。除了对角线上的图被 ist(Y(:,i))所代替之外，plotmatrix(Y)和 plotmatrix(Y,Y)是一样。

plotmatrix(...,'LineSpec')使用 LineSpec 定义的线型绘制图形。

[H,AX,BigAx,P] = plotmatrix(...)返回一个指向 H 中生成的对象的矩阵句柄，一个指向单独的子图 AX 中的对象的矩阵句柄，一个指向子图框架 BigAx 的对象的句柄，一个指向 P 中直方图对象的矩阵句柄。

举例:

绘制一组随机数的 Scatter 图。

```
x = randn(30,3); y = x*[-1 2 1;2 0 1;1 -2 3;]';
plotmatrix(y,'*r')
```

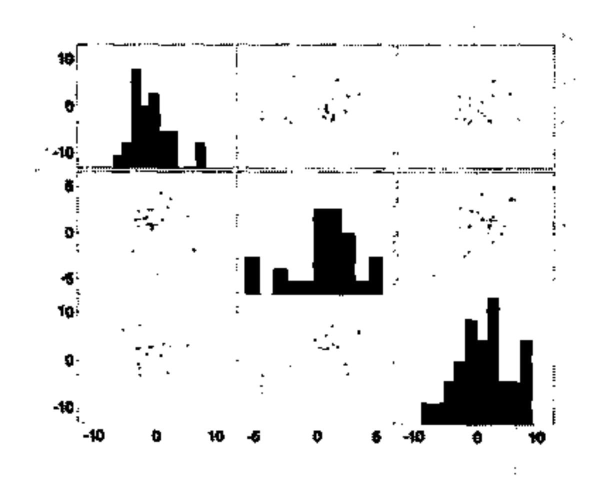


图 16-73 散点图

20. 伪色绘图

名称: pcolor

伪色绘图。

语法: 该函数有如下三种表达形式:

- pcolor(C)
- pcolor(X,Y,C)
- h = pcolor(...)

描述: 一个 pseudocolor 命令是带有由 C 确定颜色的单元的矩形排列。MATLAB 使用 C 里的每一个由四个相邻点组成的组创建一个 pseudocolor 图形来定义一个表面块(如 cell)。

pcolor(C)画出一个 pseudocolor 图形。C 的元素将一个索引线性映射到当前色图里。从 C 到当前色图的映射由 colormap 和 caxis 来定义。

pcolor(X,Y,C)在由 X 和 Y 确定的位置处画出 C 的元素的一个 pseudocolor 图形。这个图形是顶点为[X(i,j), Y(i,j)]的一个矩形二维网格。X 和 Y 是确定网格线间隔的向量或矩阵。如果 X 和 Y 是向量，X 对应 C 的列，Y 对应 C 的行。如果 X 和 Y 是矩阵，它们与 C 有相同的阶数。

h = pcolor(...)返回一个指向表面图形对象的句柄。

举例:

(1) 显示 Hadamard 矩阵的一个伪色图。

```
pcolor(hadamard(20))
colormap(gray(2))
axis ij
axis square
```

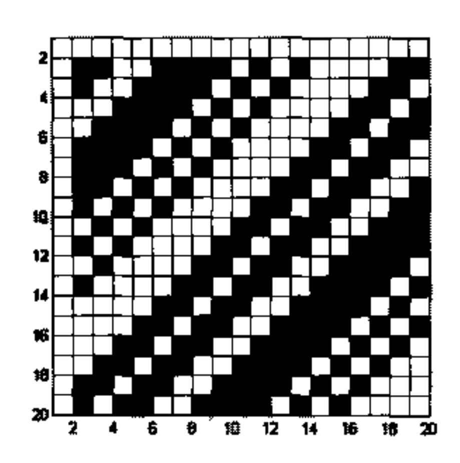


图 16-74　Hadamard 矩阵的伪色图

(2) 极坐标系下的一个简单的色轮图。

```
n = 6;
r = (0:n)'/n;
theta = pi*(-n:n)/n;
X = r*cos(theta);
Y = r*sin(theta);
C = r*cos(2*theta);
pcolor(X,Y,C)
axis equal tight
```

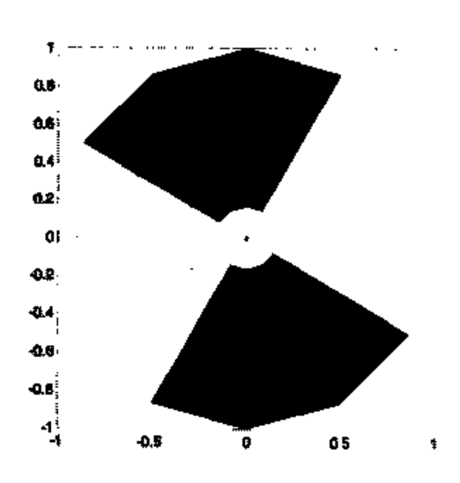


图 16-75　一个简单的色轮图

21. 极坐标直方图

名称：rose

极坐标直方图。

语法：该函数有如下四种表达形式：

- rose(theta)

- rose(theta,x)
- rose(theta,nbins)
- [tout,rout] = rose(...)

描述：rose 生成一个在极坐标中绘制的直方图。直条沿圆周分布。

rose(theta)在 20 或更少的圆周角区间上绘制 theta 角的直方图。用弧度表达的向量 theta，确定了每个直条到原点的角度。每个直条的径向长度反映了在 theta 上的频数大小。

rose(theta,x)在向量 x 指定的位置上表现 theta 的频数分布。length(x) 等于直条的数目。x 的取值则定义了每个直条的中心角度。

rose(theta,nbins)在[0. 2π]的圆周上的 nbins 个等分的区间上显示 theta 的频数分布。默认值是 20.

[tout,rout] = rose(...)返回向量 tout 和 rout，利用 polar(tout,rout) 便可以生成直方图，而此函数本身不能绘制图形。

举例：

绘制一个显示 50 个随机数分布的极坐标直方图。

theta = 2*pi*rand(1,50);

rose(theta)

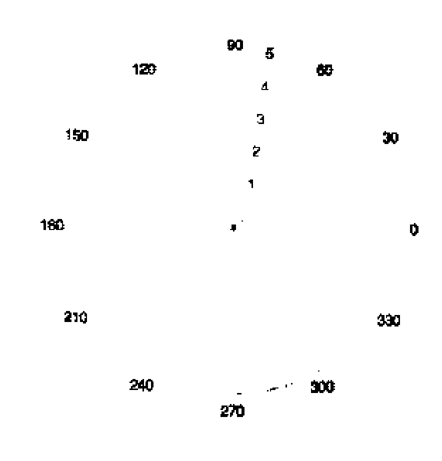


图 16-76　极坐标直方图

22. 向量场图

名称：quiver

向量场图。

语法：该函数有如下六种表达形式：

- quiver(U,V)
- quiver(X,Y,U,V)
- quiver(...,scale)
- quiver(...,LineSpec)
- quiver(...,LineSpec,'filled')
- h = quiver(...)

描述：quiver 命令在(x,y)点处显示对应于(u,v)分量的向量。

quiver(U,V)在由 x = 1:n 和 y = 1:m 定义的坐标处画出由 U 和 V 确定的向量。这里[m,n] = size(U) = size(V)。这个语法在一个矩形网格上画出 U 和 V。quiver 根据它们之间的距离自动调节向量以防止重叠。

quiver(X,Y,U,V)在每一个以 X 和 Y 中元素为元素的有序对处画出向量。如果 X 和 Y 是向量，length(X) = n 和 length(Y) = m，这里[m,n] = size(U) = size(V)，向量 X 对应 U 和 V 的列，Y 对应 U 和 V 的行。

quiver(...,scale)自动调节向量以防止重叠，然后乘以 scale。scale = 2 加倍它们相对的长度，scale = 0.5 则减半它们之间的长度。如不需要自动调节，则使用 scale = 0 来画出速度向量。

quiver(...,LineSpec)使用任何有效的 LineSpec 来确定线型、标记和颜色。quiver 在向量的起点画出这些标记。

quiver(...,LineSpec,'filled')填充由 LineSpec 确定的标记。

h = quiver(...)返回一个线句柄的向量。

举例：

画出一个函数的梯度场。

```
[X,Y] = meshgrid(-2:.2:2);
Z = X.*exp(-X.^2 - Y.^2);
[DX,DY] = gradient(Z,.2,.2);
contour(X,Y,Z)
hold on
quiver(X,Y,DX,DY)
colormap hsv
grid off
hold off
```

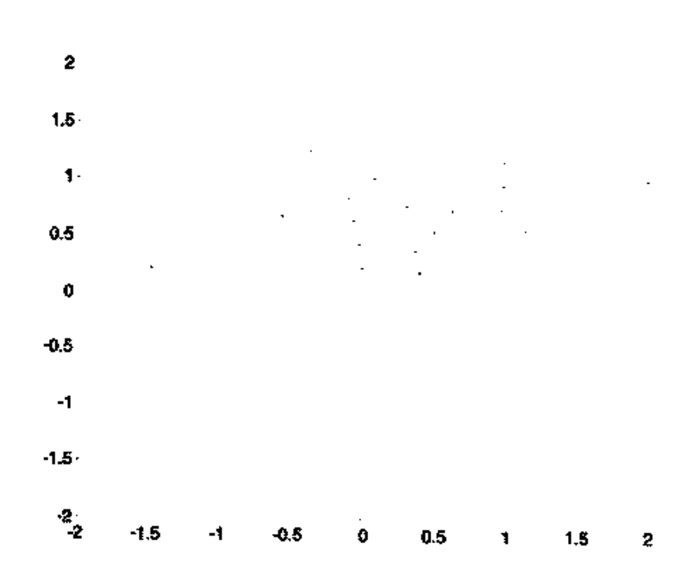


图 16-77　函数的梯度图

23. 绘制带图

名称：ribbon

绘制带图。

语法：该函数有如下四种表达形式：

- ribbon(Y)
- ribbon(X,Y)
- ribbon(X,Y,width)
- h = ribbon(...)

描述：ribbon(Y)绘制 Y 列的三维离散带图，其中的 X = 1:size(Y,1)。

ribbon(X,Y)绘制 Y-X 的三维带图。X 和 Y 是同维的向量或矩阵。

ribbon(X,Y,width)定义了带图中带的宽度。默认值是 0.75。

h = ribbon(...)返回了一个指向一个面对象的向量句柄。带图中的一个带子对应一个句柄。

举例：

生成一个 peaks 函数的带图。

```
[x,y] = meshgrid(-2:.2:2,-3:.1:3);
z = peaks(x,y);
ribbon(y,z)
colormap hsv
```

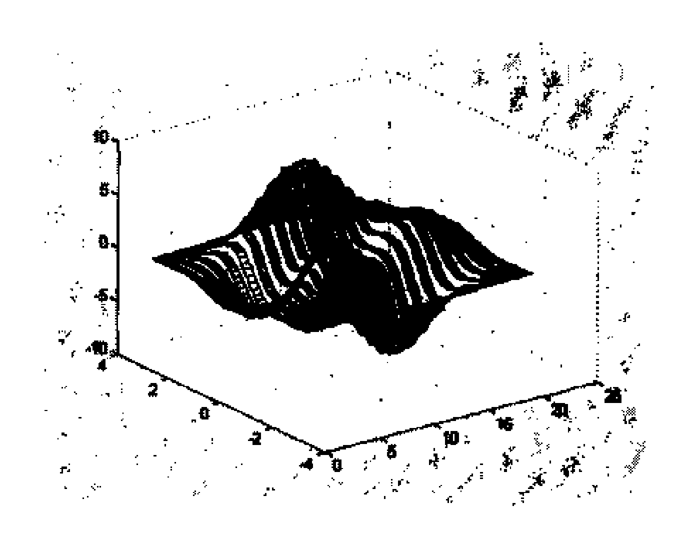


图 16-78 peaks 函数的带图

24. 阶梯曲线图

名称：stairs

阶梯曲线图。

语法：该函数有如下五种表达形式：

- stairs(Y)
- stairs(X,Y)
- stairs(...,LineSpec)

- [xb,yb] = stairs(Y)
- [xb,yb] = stairs(X,Y)

描述：Stairstep 命令用于画出数字样本数据系统的时间历史图。

stairs(Y)画出 Y 中元素的一个阶梯图。当 Y 是向量时，x 轴的范围是从 1 到 size(Y)。当 Y 是一个矩阵时，x 轴的范围是从 1 到 Y 的行数。

stairs(X,Y)画出 X 相对于 Y 的列数的图形。X 和 Y 是具有相同维数的向量或具有相同阶数的矩阵。X 可以是一个行向量或一个列向量，Y 可以是具有 length(X)行的一个矩阵。

stairs(...,LineSpec)为图形指定一个线型、标记和颜色。

[xb,yb] = stairs(Y)和[xb,yb] = stairs(x,Y)并不画出图形，而是返回向量 xb 和 yb，使得 plot(xb,yb)画出阶梯图。

举例：

创建一个余弦波的阶梯图。

```
x = 0:.25:10;
stairs(x,cos(x))
```

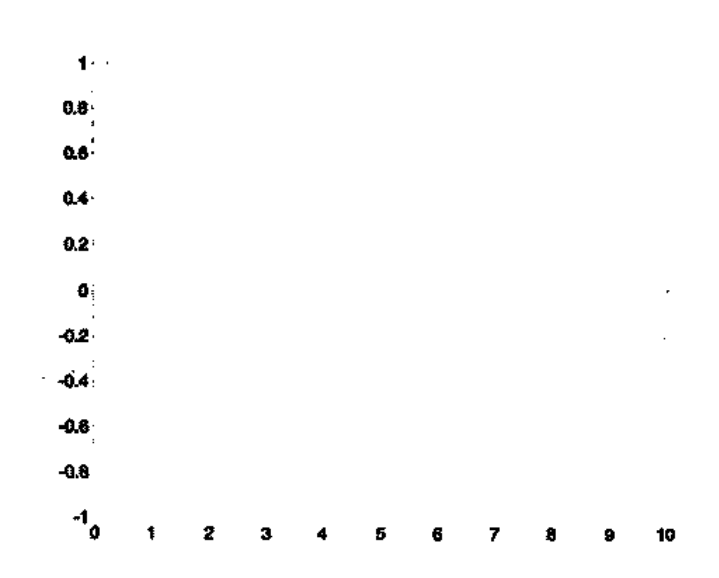


图 16-79　余弦波的阶梯图

25. 二维离散点图

名称：scatter

二维离散点图。

语法：该函数有如下六种表达形式：

- scatter(X,Y,S,C)
- scatter(X,Y)
- scatter(X,Y,S)
- scatter(...,markertype)
- scatter(...,'filled')
- h = scatter(...,)

描述：scatter(X,Y,S,C)在向量 X 和 Y 定义的位置绘制彩色的圆圈标记（X 和 Y 必须大小相同）。S 定义了每个标记符号的大小。S 可以是同 X 和 Y 同样尺寸的向量，也可以是标量。如果是标量，则 MATLAB 绘制所有的标记符号的大小相同。C 定义了每个标记的颜色。当 C 是与 X 和 Y 同维的向量时，C 的分量线性对应到当前的色图中；当 C 是 length(X)×3 的矩阵时，它定义了 RGB 的颜色矩阵。此外 C 也可以是一个颜色字符串。

scatter(X,Y)在使用默认的位置值和颜色值绘制标记。

scatter(X,Y,S)用同一颜色绘制标记，标记大小由 sizes (S)定义。

scatter(...,markertype)使用 markertype 所定义的标记符号的类型绘制点图。

scatter(...,'filled')填充每个圆圈标记。

h = scatter(...) 返回指向由 scatter 所生成的线对象的句柄。

举例：

```
load seamount
scatter(x,y,8,z)
```

-47.95
-48
-48.05
-48.1
-48.15
-48.2
-48.25
-48.3
-48.35
-48.4
-48.45
210.8 210.9 211 211.1 211.2 211.3 211.4 211.5 211.6 211.7 211.8

图 16-80 二维散点图图

26. 三维散点图

名称：scatter3

三维散点图。

语法：该函数有如下六种表达形式：

- scatter3(X,Y,Z,S,C)
- scatter3(X,Y,Z)
- scatter3(X,Y,Z,S)
- scatter3(...,markertype)
- scatter3(...,'filled')
- h = scatter3(...)

描述：scatter3(X,Y,Z,S,C)在由向量 X、Y 和 Z 确定的位置处显示带有颜色的小圆(它们必须具有同样的大小)。S 确定每一个标记的尺寸。S 可以是与 X、Y 和 Z 同样大小的一个向量，或是一个标量。如果 S 是一个标量，MATLAB 画出所有尺寸相同的标记。C 确定每一

个标记的颜色。当 C 是一个与 X、Y 和 Z 有相同长度的向量时，C 中的值线性映射于当前色图中的颜色。当 C 是一个 length(X)×3 的矩阵时，它把标记的颜色指定为 RGB 值。C 还可以是一个颜色串。

scatter3(X,Y,Z)画出默认尺寸和颜色下的标记。

scatter3(X,Y,Z,S)在指定的 sizes(S)处用一种颜色画出标记。

scatter3(...,markertype)使用指定的标记类型，而不是'o'。

scatter3(...,'filled')填充这些标记。

h = scatter3(...)返回一个指向由 scatter3 创建的线对象的句柄。

举例：

```
[x,y,z] = peaks;
X = [x(:)*.5 x(:)*.75 x(:)];
Y = [y(:)*.5 y(:)*.75 y(:)];
Z = [z(:)*.5 z(:)*.75 z(:)];
S = repmat([1 .75 .5]*10,prod(size(x)),1);
C = repmat([1 2 3],prod(size(x)),1);
scatter3(X(:),Y(:),Z(:),S(:),C(:),'filled')
```

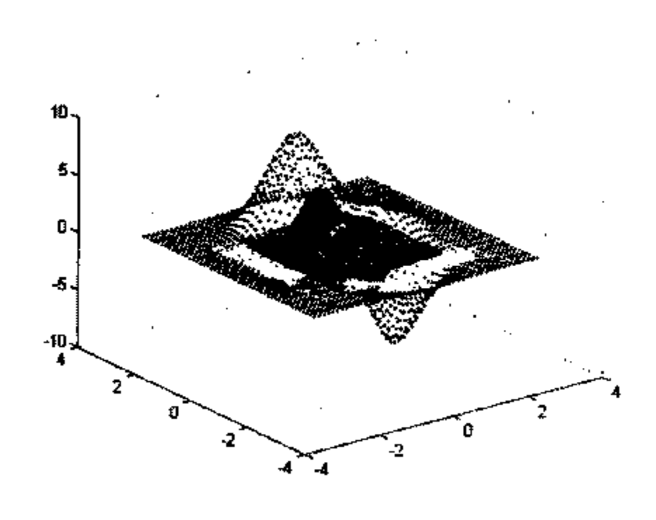


图 16-81　三维散点图

27. 绘制二维火柴杆图

名称：stem

绘制二维火柴杆图。

语法：该函数有如下五种表达形式：

- stem(Y)
- stem(X,Y)
- stem(...,'fill')
- stem(...,LineSpec)
- h = stem(...)

描述：二维 stem 函数绘制从 x 轴上伸展出的火柴杆直线图。每个火柴杆直线由一个圆圈（默认值）或其他的标记符号终止，符号在 y 轴方向上的位置代表了其数值的大小。

stem(Y)依据数据序列 Y 的大小绘制从 x 轴上伸展的火柴杆直线。当 Y 是矩阵时，stem 对 Y 的每一行的所有元素对应同样的 x 绘制火柴杆图。

stem(X,Y)绘制 X-Y 的火柴杆图，其中 X 和 Y 是同样尺寸的向量或矩阵。而且 X 可以是行向量或列向量，而 Y 相应是矩阵，其行数是 length(X)。

stem(...,'fill')指定是否在火柴杆直线末端的圆圈内填充颜色。

stem(...,LineSpec)为绘制的火柴杆直线指定线型，标记符号和颜色。

h = stem(...)返回指向火柴杆直线图形对象的句柄。

举例：

绘制 10 个随机数的火柴杆图：

```
y = linspace(0,1,8);
stem(exp(-2*y),'fill','-.')
axis ([0 11 0 1])
```

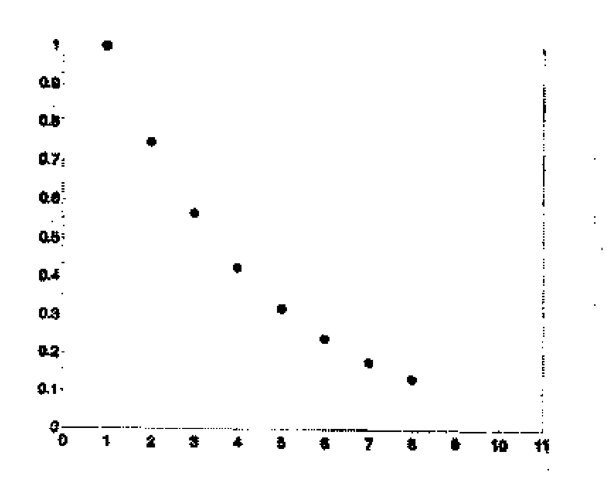


图 16-82 火柴杆图

28. 凸壳图

名称：convhull

凸壳图。

语法：该函数有如下两种表达形式：

- K = convhull(x,y)
- K = convhull(x,y,TRI)

描述：K = convhull(x,y)将索引返回到凸壳上点的 x 和 y 向量。

K = convhull(x,y,TRI)使用三角化(如从 delaunay 得到的)而不是每一次都进行计算。

举例：

```
xx = -1:.05:1; yy = abs(sqrt(xx));
[x,y] = pol2cart(xx,yy);
k = convhull(x,y);
```

```
plot(x(k),y(k),'r-',x,y,'b+')
```

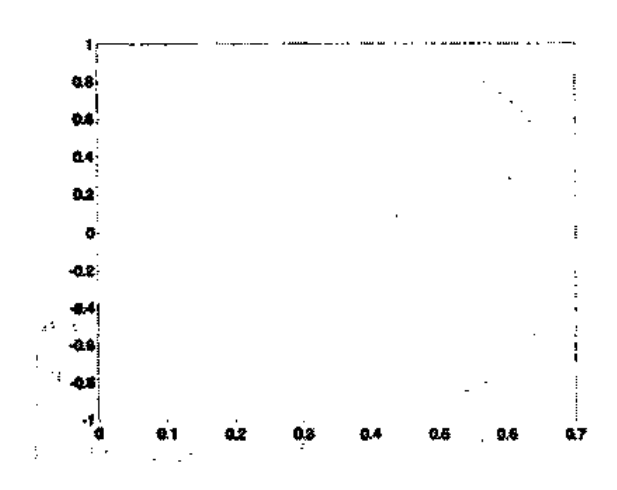


图 16-83　凸壳图

29. 检测点是否在多边形内

名称：inpolygon

检测点是否在多边形内。

语法：IN = inpolygon(X,Y,xv,yv)

描述：IN = inpolygon(X,Y,xv,yv)返回一个与 X 和 Y 同阶的矩阵 IN。它的每一个元素根据点(X(p,q),Y(p,q))是否在多边形内只从 1，0.5，0 中取值，而多边形由向量 xv 和 yv 定义。具体而言：如果(X(p,q),Y(p,q))在多边形内：IN(p,q) = 1；如果(X(p,q),Y(p,q))在多边形边界上：IN(p,q) =0.5；如果(X(p,q),Y(p,q))在多边形外：IN(p,q) = 0。

举例：

```
L = linspace(0,2.*pi,6); xv = cos(L)';yv = sin(L)';
xv = [xv ; xv(1)]; yv = [yv ; yv(1)];
x = randn(250,1); y = randn(250,1);
in = inpolygon(x,y,xv,yv);
plot(xv,yv,x(in),y(in),'r+',x(~in),y(~in),'bo')
```

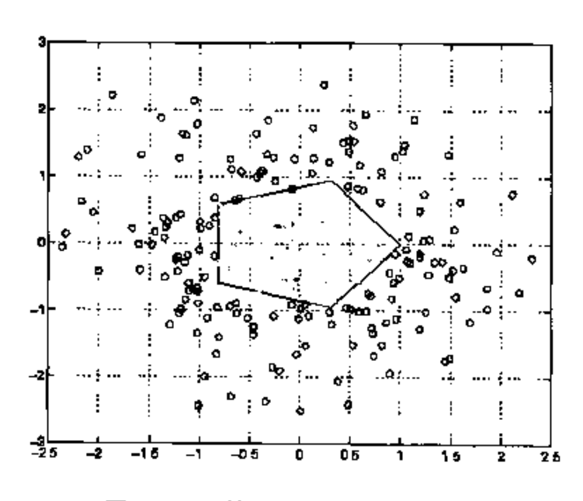


图 16-84　检查点是否在多边形内

30. 搜索最近点

名称：dsearch

搜索最近点。

语法：该函数有如下两种表达形式：

- K = dsearch(x,y,TRI,xi,yi)
- K = dsearch(x,y,TRI,xi,yi,S)

描述： K = dsearch(x,y,TRI,xi,yi)将最近的点(x,y)的索引返回到点(xi,yi)。dsearch 要求从 delaunay 得到的点 x，y 的一个三角化 TRI。

K = dsearch(x, y, TRI, xi, yi, S)使用的是稀疏矩阵 S 而不是每一次都计算它：S = sparse(TRI(:,[1 1 2 2 3 3]),TRI(:,[2 3 1 3 1 2]),1,nxy,nxy)，这里 nxy = prod(size(x))。

31. 多边形的面积

名称：polyarea

多边形的面积。

语法：该函数有如下两种表达形式：

- A = polyarea(X,Y)
- A = polyarea(X,Y,dim)

描述：A = polyarea(X,Y)返回由向量 X 和 Y 内的顶点确定的多边形的面积。如果 X 和 Y 是相同阶数的矩阵，则 polyarea 返回由 X 和 Y 的列定义的多边形的面积。

如果 X 和 Y 是多维数组，polyarea 返回在 X 和 Y 里的多边形的面积。

A = polyarea(X,Y,dim)沿着由标量 dim 确定的尺寸进行操作。

举例：

```
L = linspace(0,2.*pi,6); xv = cos(L)';yv = sin(L)';
xv = [xv ; xv(1)]; yv = [yv ; yv(1)];
A = polyarea(xv,yv);
plot(xv,yv); title(['Area = ' num2str(A)]); axis image
```

Area = 2.3776

0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8

-0.5 0 0.5 1

图 16-85 多边形及其面积图

32. Voronoi 图

名称：voronoi

Voronoi 图。

语法：该函数有如下四种表达形式：

- voronoi(x,y)
- voronoi(x,y,TRI)
- h = voronoi(...,'LineSpec')
- [vx,vy] = voronoi(...)

描述：voronoi(x,y)为点 x、y 画出 Voronoi 图。

voronoi(x,y,TRI)使用三角化 TRI 而不是通过 delaunay 计算它。

h = voronoi(...,'LineSpec')使用确定的颜色和线型画出图形，并返回指向在 h 里创建的线对象的句柄。

[vx,vy] = voronoi(...)在 vx 和 vy 里返回 Voronoi 边的顶点，使得 plot(vx,vy,'-',x,y,'.')创建 Voronoi 图形。

举例：

本例对 10 个随机生成的点画出 Voronoi 图。

```
rand('state',0);
x = rand(1,10); y = rand(1,10);
[vx, vy] = voronoi(x,y);
plot(x,y,'r+',vx,vy,'b-'); axis equal
```

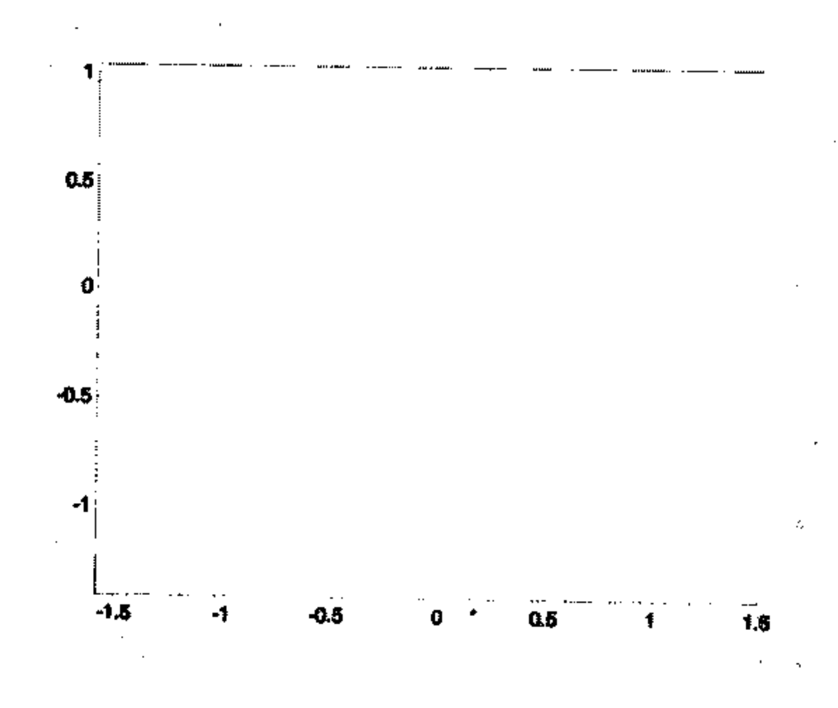


图 16-86　Voronoi 图

16.8 视角控制

1. 移动照相机的位置和目标

名称：camdolly

移动照相机的位置和目标。

语法：该函数有如下四种表达形式：

- camdolly(dx,dy,dz)
- camdolly(dx,dy,dz,'targetmode')
- camdolly(dx,dy,dz,'targetmode','coordsys')
- camdolly(axes_handle,...)

描述：camdolly 根据指定的数量移动照相机的位置和照相机的目标。

camdolly(dx,dy,dz)根据指定的数量。移动照相机的位置和照相机的目标。

camdolly(dx,dy,dz,'targetmode')targetmode 参数能打开二个参数，以决定 MATLAB 如何移动照相机：movetarget (默认值)——同时移动照相机和目标；fixtarget——只移动照相机。

camdolly(dx,dy,dz,'targetmode','coordsys')中，coordsys 参数打开三个参数，以决定MATLAB 如何理解 dx、dy 和 dz：

坐标系：camera (默认值)——在照相机的坐标系中移动，dx 决定左右移动，dy 决定上下移动，dz 沿观测轴移动。所有的单位与 The units are normalized to the scene；pixels——把 dx 和 dy 看作像素的偏移量，且忽略 dz；data——把 dx, dy,和 dz 看作坐标轴上坐标的偏移量。

camdolly(axes_handle,...)在由第一个参数 axes_handle 确定的轴上移动。如果没有定义轴句柄，则在当前轴上移动。

举例：

此例把照相机在 x 和 y 轴进行一系列的移动。

```
surf(peaks)
axis vis3d
t = 0:pi/20:2*pi;
dx = sin(t)./40;
dy = cos(t)./40;
for i = 1:length(t);
    camdolly(dx(i),dy(i),0)
    drawnow
end
```

2. 确定照相机的位置来观察对象

名称：camlookat

确定照相机的位置来观察一个对象或一组对象。

语法：该函数有如下三种表达形式：

- camlookat(object_handles)
- camlookat(axes_handle)

- camlookat

描述：camlookat(object_handles)观察在向量对象句柄里确定的对象。这个向量可以包括轴的附属对象的句柄。

camlookat(axes_handle)查看那些是由轴句柄确定的轴的附属对象。

camlookat 观察那些在当前轴里的对象。

举例：

本例在不同的位置处创建三个球面，并且变化照相机的位置来显示球面。

```
[x y z] = sphere;
s1 = surf(x,y,z);
hold on
s2 = surf(x+3,y,z+3);
s3 = surf(x,y,z+6);
dasp ect([1 1 1])
view(30,10)
camproj perspective
caml ookat(gca) % Compose the scene around the current axes
pause(2)
camlookat(s1)   % Compose the scene around sphere s1
pause(2)
camlookat(s2)   % Compose the scene around sphere s2
pause(2)
camlookat(s3)   % Compose the scene around sphere s3
pause(2)
camlookat(gca)
```

3. 绕照相机的目标旋转照相机

名称：camorbit

绕照相机的目标旋转照相机。

语法：该函数有如下四种表达形式

- camorbit(dtheta,dphi)
- camorbit(dtheta,dphi,'coordsys')
- camorbit(dtheta,dphi,'coordsys','direction')
- camorbit(axes_handle,...)

描述：camorbit(dtheta,dphi)绕照相机的目标旋转照相机，旋转的角度由 dtheta 和 dphi 确定(单位都是度)其中，dtheta 是水平旋转角度，dphi 是垂直旋转角度。

camorbit(dtheta,dphi,'coordsys')中的参数 coordsys 决定旋转的中心，可以取两个值：data (默认值)——绕由目标定义的轴的方向旋转（默认值是 z 轴正方向）；camera——绕目标定义的点旋转。

camorbit(dtheta,dphi,'coordsys','direction')中的 direction 参数，连同目标，定义了相对于数据坐标系的旋转轴，旋转方向由 x，y，z 为分量的三维向量来确定。

camorbit(axes_handle,...)绕第一个参数 axes_handle 所指定的轴旋转。当用户没有指定一个轴句柄时，便取当前轴。

举例：

比较 for 循环中的在两个坐标系中的旋转。

```
surf(peaks)
axis vis3d
for i=1:36
    camorbit(10,0,'data',[0 1 0])
    drawnow
end
surf(peaks)
axis vis3d
for i=1:36
    camorbit(10,0,'camera')
    drawnow
end
```

4. 围绕照相机的位置旋转照相机目标

名称：campan

围绕照相机的位置旋转照相机目标。

语法：该函数有如下四种表达形式：

- campan(dtheta,dphi)
- campan(dtheta,dphi,'coordsys')
- campan(dtheta,dphi,'coordsys','direction')
- campan(axes_handle,...)

描述：campan(dtheta,dphi)围绕照相机的位置旋转照相机的目标，其旋转量由 dtheta 和 dphi 决定(以度为单位)。dtheta 是水平旋转，dphi 是垂直旋转。

campan(dtheta,dphi,'coordsys')中 coordsys 选项确定旋转的中心。它有两个值：data (默认)——围绕由照相机位置和方向(默认是正 z 方向)定义的一个轴，旋转照相机的目标；camera——关于由照相机的目标定义的点旋转照相机。

campan(dtheta,dphi,'coordsys','direction')的 direction 选项，联合照相机位置，为数据坐标系统定义了旋转轴。将 direction 指定为一个包含了方向的 x、y 和 z 分量的三元素向量，或 x、y 和 z 本身，其中 x，y 或 z 分别用于显示[1 0 0]，[0 1 0]或[0 0 1]。

campan(axes_handle,...)在由第一个选项 axes_handle 确定的轴上进行操作。当用户不用指定一个轴句柄时，campan 在当前轴上进行操作。

5. 设置或查询照相机的位置

名称：campos

设置或查询照相机的位置。

语法：该函数有如下六种表达形式：

- campos

- campos([camera_position])
- campos('mode')
- campos('auto'
- campos('manual')
- campos(axes_handle,...)

描述：无参数的 campos 返回照相机在当前图形中的位置。

campos([camera_position])设置照相机在当前图形中的位置。指定的位置由含有 x，y，z 坐标的三维向量确定。

campos('mode')返回照相机的位置状态：auto 或 manual。

campos('auto')设置照相机的位置状态为 auto。

campos('manual')设置照相机的位置状态为 manual。

campos(axes_handle,...)设置或查询由第一个参数 axes_handle 所指向的轴的照相机的位置状态。如果没有指定，则为当前轴。

举例：

此例沿 x 轴方向一步步移动照相机：

```
surf(peaks)
axis vis3d off
for x = -200:5:200
    campos([x,5,10])
    drawnow
end
```

6. 设置或查询投影类型

名称：camproj

设置或查询投影类型。

语法：该函数有如下三种表达形式：

- camproj
- camproj(projection_type)
- camproj(axes_handle,...)

描述：projection type 确定 MATLAB 是否对三维视图使用透视投影或正交投影。没有选项的 camproj 返回设置在当前轴里的投影类型。

camproj('projection_type')当前轴里将投影类型设置为确定的值。projection_type 的可能值为：orthographic 和 perspective。

camproj(axes_handle,...)在由第一选项 axes_handle 确定的轴上执行设置或查询。当用户不想指定一个轴句柄时，camproj 在当前轴上进行操作。

7. 绕视轴旋转照相机

名称：camroll

绕视轴旋转照相机。

语法：该函数有如下两种表达形式：

- camroll(dtheta)

- camroll(axes_handle,dtheta)

描述：camroll(dtheta)绕照相机的视轴旋转 dtheta 所指定角度。视轴指从照相机到目标的连线。

camroll(axes_handle,dtheta)在由 axes_handle 所指定的轴上操纵照相机的旋转。如果没有指定对象，则对当前轴进行操纵。

8. 设置或查询照相机目标的位置

名称：camtarget

设置或查询照相机目标的位置。

语法：该函数有如下六种表达形式：

- camtarget
- camtarget([camera_target])
- camtarget('mode')
- camtarget('auto')
- camtarget('manual')
- camtarget(axes_handle,...)

描述：照相机目标是照相机所指的位置。照相机指向这个点的方向，而忽略其位置。没有选项的 camtarget 返回当前轴里照相机目标的位置。

camtarget([camera_target])在当前轴里将照相机目标设置为确定的值。在轴的数据单元里，将目标确定含 x、y 和 z 坐标的一个三元素向量。

camtarget('mode')返回照相机目标模式的值，这个值或者是 auto(默认)，或者是 manual。

camtarget('auto')将照相机目标模式设置为 auto；camtarget('manual') 将照相机目标模式设置为 manual。

camtarget(axes_handle,...)在由第一选项 axes_handle 确定的轴上执行设置或询问。当用户不想指定一个轴句柄时，camtarget 在当前轴上进行操作。

举例：

本例沿 x 轴移动照相机的位置和照相机目标。

```
surf(peaks);
axis vis3d
xp = linspace(-150,40,50);
xt = linspace(25,50,50);
for i=1:50
      campos([xp(i),25,5]);
      camtarget([xt(i),30,0])
      drawnow
end
```

9. 设置或查询照相机视角

名称：camva

设置或查询照相机视角。

语法：该函数有如下六种表达形式：

- camva
- camva(view_angle)
- camva('mode')
- camva('auto')
- camva('manual')
- camva(axes_handle,...)

描述：照相机的视角决定了照相机的视图区。较大的角度产生较小的景物视图。用户可以通过改变照相机的视角来实现缩放功能。没有选项的 camva 返回设置在当前轴里的照相机视角。

camva(view_angle)在当前轴里将 view_angle 设置为确定的值。用度来确定视角。

camva('mode')返回照相机视角模式的当前值，这个值或者是 auto(默认)，或者是 manual。

camva('auto')将照相机的视角模式设置为 auto；camva('manual') 将照相机的视角模式设置为 manual。

camva(axes_handle,...)在由第一选项 axes_handle 确定的轴上执行设置或询问。当用户不想指定一个轴句柄时，camva 在当前轴上进行操作。

举例：

本例创建了两个按钮，一个用于放大，一个用于缩小。

```
uicontrol('Style','pushbutton',...
   'String','Zoom In',...
   'Position',[20 20 60 20],...
   'Callback','if camva <= 1;return;else;camva(camva-1);end');
uicontrol('Style','pushbutton',...
   'String','Zoom Out',...
   'Position',[100 20 60 20],...
   'Callback','if camva >= 179;return;else;camva(camva+1);end');
```

现在创建一个图形来放大和缩小：

```
surf(peaks);
```

注意视角的取值范围为 0 到 180 度。

10. 设置或查询照相机方向

名称：camup

设置或查询照相机方向。

语法：该函数有如下六种表达形式：

- camup
- camup([up_vector])
- camup('mode')
- camup('auto')
- camup('manual')
- camup(axes_handle,...)

描述：无参数的 camup 返回当前设置下的照相机的方向。

camup([up_vector])设置在当前图形中照相机的方向指向。方向向量有 x、y、z 三个分量确定。

camup('mode') 返回当前照相机方向向量的控制模式：auto（默认值）或 manual。

camup('auto')设置照相机的方向向量的控制模式为 auto。在 auto 模式下，MATLAB 给定的方向向量的默认值是[0 1 0]，即二维视图。

camup('manual')设置照相机的方向向量的控制模式为 manual。在 manual 模式下，MATLAB 保持当前的方向向量值。

camup(axes_handle,...)设置或查询由参数 axes_handle 所指定的图形上的照相机的方向向量。如果没有指定，则对当前的图形对象执行设置或查询。

11. 放大和缩小

名称：camzoom

放大和缩小。

语法：该函数有如下两种表达形式：

- camzoom(zoom_factor)
- camzoom(axes_handle,...)

描述：camzoom(zoom_factor) 对屏幕上的图形进行放大或缩小。如果缩放比例系数 zoom_factor 大于 1，则放大；反之，缩小。

camzoom(axes_handle,...)对参数 axes_handle 所指向的图形进行缩放。如果没有指定，则对当前图形进行缩放。

12. 设置或查询轴的纵横比

名称：daspect

设置或查询轴的纵横比。

语法：该函数有如下六种表达形式：

- daspect
- daspect([aspect_ratio])
- daspect('mode')
- daspect('auto')
- daspect('manual')
- daspect(axes_handle,...)

描述：数据的 aspect ratio 确定沿 x、y 和 z 轴的数据单位的相对刻度。没有选项的 daspect 返回当前轴的纵横比。

daspect([aspect_ratio])将当前轴里的数据形状设置为确定的值。将纵横比确定为表示 x、y 和 z 轴刻度比值的三个相对的值([1 1 3]意味着 x 轴的一个单位在长度上等于 y 轴的一个单位和 z 轴的三个单位)。

daspect('mode')返回纵横比的当前值，这个值或者是 auto(默认)，或者是 manual。

daspect('auto')将纵横比设置为 auto；daspect('manual') 将纵横比设置为 manual。

daspect(axes_handle,...)在由第一选项 axes_handle 确定的轴上执行设置或询问。当用户不想指定一个轴句柄时，daspect 在当前轴上进行操作。

举例：

(1) 下面的一个函数 $z = x*exp(-x^2-y^2)$的表面图可用于阐明数据的纵横比。

首先在区间 $-2 \le x \le 2, -2 \le y \le 2$ 上画出函数图：

```
[x,y] = meshgrid([-2:.2:2]);
z = x.*exp(-x.^2 - y.^2);
surf(x,y,z)
```

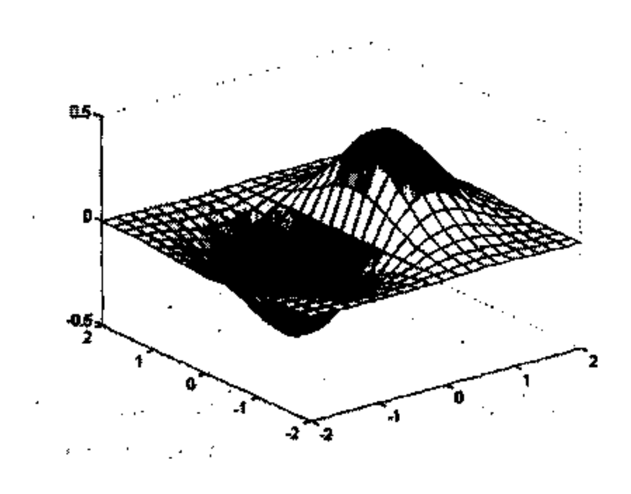


图 16-87 图形的纵横比显示

查询数据形状比：

```
daspect
```

其输出结果为：

```
ans =
     4   4   1
```

(2) 将数据纵横比设置为[1 1 1]来产生等刻度轴的一个表面图。

```
daspect([1 1 1])
```

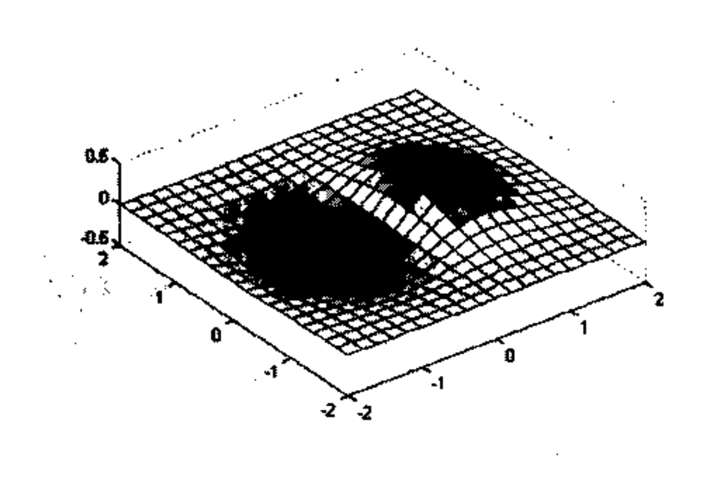


图 16-88 设置纵横比的表面图

13. 设置或查询绘图框的纵横比

名称：pbaspect

设置或查询绘图框的纵横比。

语法：该函数有如下六种表达形式：

- pbaspect
- pbaspect([aspect_ratio])
- pbaspect('mode')
- pbaspect('auto')
- pbaspect('manual')
- pbaspect(axes_handle,...)

描述：绘图框的纵横比决定了 x-, y-, 和 z-轴的相对大小。无参数的 pbaspect 返回当前图形窗口的绘图框的纵横比。

pbaspect([aspect_ratio])设置当前图形中的绘图框的纵横比为指定值。如：[1 1 1] (默认值)，意味着绘图框成正方体。

pbaspect('mode')返回当前绘图框的纵横比的控制模式：auto（默认值）或 manual。

pbaspect('auto')设置绘图框的纵横比的控制模式为 auto；pbaspect('manual')设置绘图框的纵横比的控制模式为 manual。

pbaspect(axes_handle,...)对参数 axes_handle 所指向的图形设置或查询绘图框的纵横比。如果没有指定，则对当前图形对象进行设置或查询。

举例：

下面的例子图解绘图框纵横比的用途。首先在-2≤x≤2, -2≤y≤2 范围内绘制曲面图：

```
[x,y] = meshgrid([-2:.2:2]);
z = x.*exp(-x.^2 - y.^2);
surf(x,y,z)
```

查询绘图框的纵横比，可见是正方体。

```
pbaspect
ans =
     1   1   1
```

同时可查询其数据纵横比：

```
daspect
ans =
     4   4   1
```

为了了解绘图框纵横比和数据纵横比的相互作用，设置数据纵横比为[1 1 1]，然后在查询绘图框纵横比。

```
daspect([1 1 1])
pbaspect
ans =
     4   4   1
```

可见绘图框纵横比调整到与数据纵横比相适应的数值。现在可以用户同时设置绘图框

的纵横比也为[1 1 1]：

pbaspect([1 1 1])

可以观察两者的对照效果：

```
upper_plot = subplot(211);
surf(x,y,z)
lower_plot = subplot(212);
surf(x,y,z)
pbaspect(upper_plot,'manual')
```

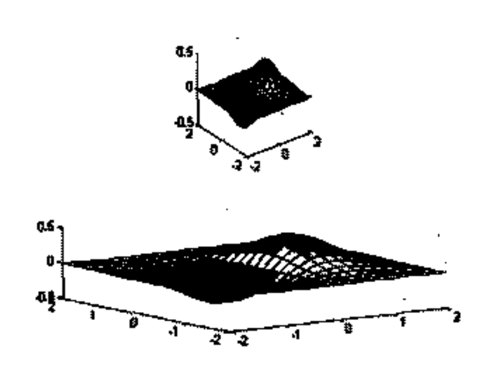


图 16-89 纵横比的用途图

14. 确定视点

名称：view

确定视点。

语法：该函数有如下八种表达形式：

- view(az,el)
- view([az,el])
- view([x,y,z])
- view(2)
- view(3)
- view(T)
- [az,el] = view
- T = view

描述：观察者的位置(视点)确定了轴的方位。用户根据方位角和仰角或通过三维空间里的一点来确定视点。

view(az,el)和 view([az,el])为一个三维图形设置视角。方位角 az 是用度来度量的从负 y 轴开始的绕 z 轴的水平旋转。正值是指观测点的逆时针旋转。el 是观测点的垂直仰角。仰角的正值对应对象上方的移动，负值对应对象下方的移动。

view([x,y,z])用笛卡尔坐标系设置视点。(x,y,z)的量值是忽略的。

view(2)设置默认的二维视图，az = 0, el = 90；view(3)设置默认的三维视图，az = --37.5, el = 30。

view(T)根据变换矩阵 T 设置视图，T 是 4×4 的矩阵，例如，由 viewmtx 生成的透视变换。

[az,el] = view 返回当前的方位角和仰角。

T = view 返回当前的 4x4 变换矩阵。

举例：

从正上方观察对象：

```
az = 0;
el = 90;
view(az, el);
```

绕 z 轴旋转 180 度：

```
az = 180;
el = 90;
view(az, el);
```

15. 视角变换矩阵

名称：viewmtx

视角变换矩阵。

语法：该函数有如下三种表达形式：

- T = viewmtx(az,el)
- T = viewmtx(az,el,phi)
- T = viewmtx(az,el,phi,xc)

描述：viewmtx 计算一个 4×4 的正交或透视变换矩阵，该矩阵用来把一个四维的向量投影到二维视平面上（如用户的计算机屏幕）。

T = viewmtx(az,el)返回一个对应方位角 az 和仰角 el 的正交变换矩阵。az 是视点的水平旋转角，而 el 是视点的仰角。该矩阵同下面的指令返回的矩阵相同：view(az,el)，T = view 但不改变当前的视角。

T = viewmtx(az,el,phi)返回一个投影矩阵。phi 是投影角。phi 控制了透视效果。

表 16-2 **投影角表**

Phi	描述
0°	正交投影
10°	类似长焦镜头
25°	类似普通镜头
60°	类似广角镜头

用户可以利用命令 view(T)返回的矩阵设置视角变换矩阵。

T = viewmtx(az,el,phi,xc)对目标点 xc 返回投影矩阵。xc 是视角的所对准的中心点。用户可以对应 xc 为三维向量 xc = [xc,yc,zc]。默认值是 xc = [0,0,0]。

举例：

确定对三维数据点(0.5,0.0,--3.0)的二维投影向量，使用默认的视角。

```
A = viewmtx(--37.5,30);
x4d = [.5   0   --3   1]';
```

```
x2d = A*x4d;
x2d = x2d(1:2)
x2d =
     0.3967
    --2.4459
x = [0  1  1  0  0  0  1  1  0  0  1  1  1  1  0  0];
y = [0  0  1  1  0  0  0  1  1  0  0  0  1  1  1  1];
z = [0  0  0  0  0  1  1  1  1  1  1  0  0  1  1  0];
```

绘制出其图像：

```
A = viewmtx(--37.5,30);
[m,n] = size(x);
x4d = [x(:),y(:),z(:),ones(m*n,1)]';
x2d = A*x4d;
x2 = zeros(m,n); y2 = zeros(m,n);
x2(:) = x2d(1,:);
y2(:) = x2d(2,:);
plot(x2,y2)
```

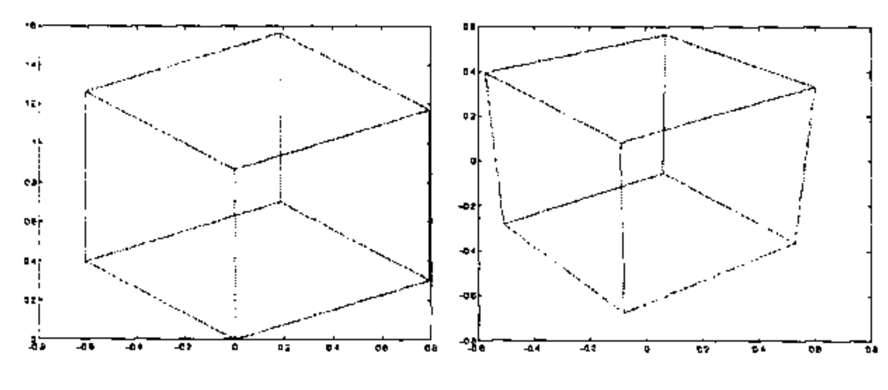


图 16-90 投影图

对其进行投影，视角为 25 度：

```
A = viewmtx(--37.5,30,25);
x4d = [.5  0  --3  1]';
x2d = A*x4d;
x2d = x2d(1:2)/x2d(4)     % Normalize
x2d =
     0.1777
    --1.8858
```

绘图：

```
A = viewmtx(--37.5,30,25);
[m,n] = size(x);
x4d = [x(:),y(:),z(:),ones(m*n,1)]';
```

```
x2d = A*x4d;
x2 = zeros(m,n); y2 = zeros(m,n);
x2(:) = x2d(1,:)./x2d(4,:);
y2(:) = x2d(2,:)./x2d(4,:);
plot(x2,y2)
```

16. 设置或查询轴的刻度范围

名称：xlim, ylim, zlim

设置或查询轴的刻度范围。

语法：该函数有如下六种表达形式：

- xlim
- xlim([xmin xmax])
- xlim('mode')
- xlim('auto')
- xlim('manual')
- xlim(axes_handle,...)

描述：没有选项的 xlim 返回当前轴各自的刻度范围。

xlim([xmin xmax])将当前轴的刻度范围设置为确定的值。

xlim('mode')返回轴刻度范围模式的当前值，这个值或者是 auto(默认)，或者是 manual。

xlim('auto')设置轴刻度范围模式为 auto。

xlim('manual')设置轴刻度范围模式为 manual。

xlim(axes_handle,...)在由第一选项 axes_handle 确定的轴上执行设置或询问。当用户不想指定一个轴句柄时，这些函数在当前轴上进行操作。

举例：

本例阐明了如何设置 x 轴和 y 轴的刻度范围来匹配数据。

```
[x,y] = meshgrid([-1.75:.2:3.25]);
z = x.*exp(-x.^2-y.^2);
surf(x,y,z)
xlim([-1.75 3.25])
ylim([-1.75 3.25])
```

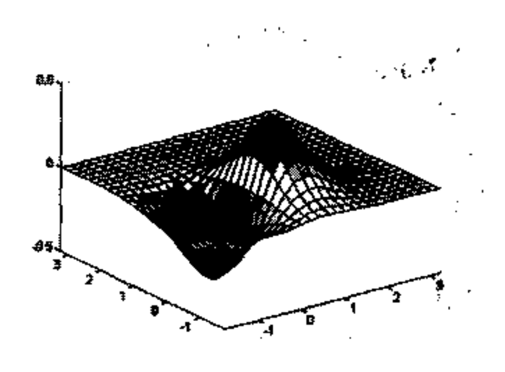


图 16-91　坐标匹配图

17. 在照相机坐标系中生成或移动光源体

名称：camlight

在照相机坐标系中生成或移动光源体。

语法：该函数有如下八种表达形式：

- camlight headlight
- camlight right
- camlight left
- camlight
- camlight(az,el)
- camlight(...`style')
- camlight(light_handle,...)
- light_handle = camlight(...)

描述：camlight('headlight')在照相机的位置处放置一个光源。

camlight('right')在照相机的右上角放置一个光源。camlight('left') 在照相机的左上角放置一个光源。无参数的 camlight 等同于 camlight('right')。

camlight(az,el)在指定的相对于照相机位置的方位角 az 和仰角 el 处放置一个光源。照相机的目标在旋转的中心。az 和 el 的单位是角度。

camlight(...,'style')中的 style 参数有两种值：local (默认值)——点光源，光源向各个方向发光。infinite——平行光，反射平行光。

camlight(light_handle,...)使用 light_handle 所指定的光源。

light_handle = camlight(...)返回指向光源对象的句柄。

举例：

下例在照相机的左上方放置一个光源，然后光源随着照相机移动：

```
surf(peaks)
axis vis3d
h = camlight('left');
for i = 1:20;
    camorbit(10,0)
    camlight(h, 'left')
    drawnow;
end
```

18. 在球坐标系里创建或定位一个照明对象

名称：lightangle

在球坐标系里创建或定位一个照明对象。

语法：该函数有如下四种表达形式：

- lightangle(az,el)
- light_handle = lightangle(az,el)
- lightangle(light_handle,az,el)
- [ax el] = lightangle(light_handle)

描述：lightangle(az,el)在由方位角和仰角确定的位置进行照明。az 是方位角(水平)旋转，el 是垂直仰角(二者都用度表示)。关于方位角和仰角的解释与 view 命令中的相同。

light_handle = lightangle(az,el)创建一个照明位置并且在 light_handle 里返回 light 的句柄。

lightangle(light_handle,az,el)设置由 light_handle 确定的照明的位置。

[az,el] = lightangle(light_handle)返回由 light_handle 确定的照明位置的方位角和仰角。

举例：

```
surf(sphere)
axis vis3d
h = light;
for az = -50:10:50
    lightangle(h,az,30)
    drawnow
end
```

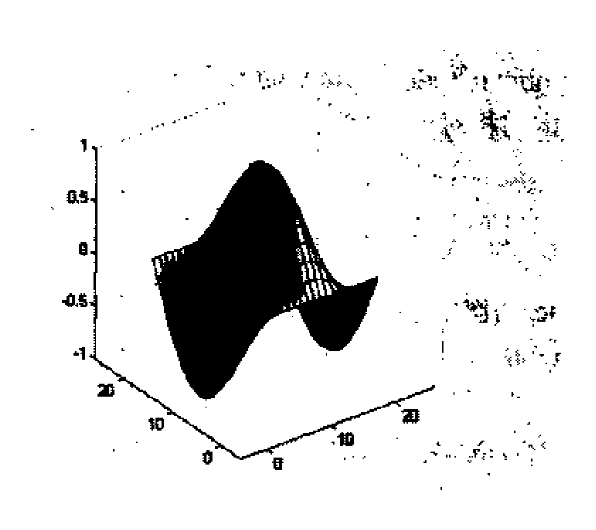


图 16-92 照明图

19. 选择照明算法

名称：lighting

选择照明算法。

语法：该函数有如下四种表达形式：

- lighting flat
- lighting gouraud
- lighting phong
- lighting none

描述：lighting 为当前图形中的光源选择其光照效果：

lighting flat 选择顶光；lighting gouraund 选择 gouraud 照明；lighting phong 选择 phong 照明；lighting none 关闭光源。

20. 控制面和块的反射比属性

名称：material

控制面和块的反射比属性。

语法：该函数有如下七种表达形式：

- material shiny
- material dull
- material metal
- material([ka kd ks])
- material([ka kd ks n])
- material([ka kd ks n sc])
- material default

描述：material 设置面对象和块对象的照明特征。

material shiny 设置反射比属性，使得对象相对于背景的明暗和颜色有很高的镜像反射

material dull 设置反射比属性，因此对象反射出更弥散的光，它没有镜子的加亮区，但是具有仅依赖于光源的反射光的颜色。

material metal 设置反射比属性，使得对象具有很高的镜像反射比和很低的背景和漫射反射比，对象还具有同时依赖于光源颜色和对象颜色的反射光的颜色。

material([ka kd ks])设置对象的背景/漫射/镜像光的强度

material([ka kd ks n])设置对象的背景/漫射/镜像光的强度和镜像指数

material([ka kd ks n sc])设置对象的背景/漫射/镜像光的强度、镜像指数和镜像颜色反射比。

material default 将对象的背景/漫射/镜像光的强度、镜像指数和镜像颜色反射比设置为它们的默认值。

16.10 颜色操作

1. 控制色图的明暗

名称：brighten

控制色图的明暗。

语法：该函数有如下四种表达形式：

- brighten(beta)
- brighten(h,beta)
- newmap = brighten(beta)
- newmap = brighten(cmap,beta)

描述：brighten 增加或减小色图的色彩强度。如果 $0 < beta < 1$，则增加色图的强度；如果 $-1 < beta < 0$，减小色图的强度。

brighten(beta)增强或减弱当前的色图。

brighten(h,beta)增强所有句柄 h 指向的对象的色彩强度。

newmap = brighten(beta)返回一个比当前色图增强或减弱的新的色图，但不影响显示的效果。

newmap = brighten(cmap,beta)返回一个比当前色图增强或减弱的新的色图，但不影响显示的效果。

举例：

增强然后减弱当前的色图。

beta = .5; brighten(beta);

beta = --.5; brighten(beta);

2. 色轴刻度

名称：caxis

色轴刻度。

语法：该函数有如下五种表达形式：

- caxis([cmin cmax])
- caxis auto
- caxis manual
- caxis(caxis)
- v = caxis

描述：caxis 控制着从数据值到色图的映射。它通过将被变址的 CData 和 CDataMapping 设置为 scaled，影响着任何的表面、块和图像。它不能通过真色 CData 或将 CDataMapping 设置为 direct 来影响表面、块和图像。

caxis([cmin cmax])将颜色的刻度范围设置为指定的最小值和最大值。小于 cmin 或大于 cmax 的数据值分别映射为 cmin 和 cmax。在 cmin 和 cmax 之间的值线性映射到当前色图上。

caxis auto 让 MATLAB 使用最小和最大数据值自动计算颜色范围。这是 MATLAB 的默认行为。颜色值将 Inf 图设置为最大颜色，将-Inf 图设置为最小颜色。不画出设置为 NaN 的颜色的面和边。

caxis manual 和 caxis(caxis)凝固标有当前范围的色轴。当保留开关打开时，它可以使后续的图形继续使用这个相同的颜色范围。

v = caxis 返回包含当前正在使用的[cmin cmax]的一个两元素行向量。

举例：

创建一个球面的(X,Y,Z)数据并将它们作为一个面来观察。

[X,Y,Z] = sphere;

C = Z;

surf(X,Y,Z,C)

C 值的范围是[-1 1]。接近-1 的 C 的值在色图里被赋为最小值，接近 1 的 C 的值在色图里被赋为最大值。

为了将表面的顶端映射为颜色表里的最高的值，使用 caxis([-1 0])

若仅使用颜色表里的最低值，键入 caxis([-1 3])

它将最低的CData值映射为色图的最低值，将最高值映射为色图的中间值。命令caxis auto 重新设置了回到 auto-ranging 的轴的刻度，并且用户在面里可以看到所有的颜色。在这种情况下，键入 caxis，返回[-1 1]。

当使用带刻度颜色数据的图像时，调节色轴是有用的。例如，为一个海角编码加载图

像数据和色图：

```
load cape
image(X,'CDataMapping','scaled')
colormap(map)
```



图 16-93　加色的一个海角图

MATLAB 设置颜色的范围来横跨图像数据，它是从 1 to 192：

```
caxis
ans =
     1     192
```

3. 画色轴

名称：colorbar

画色轴。

语法：该函数有如下五种表达形式：

- colorbar
- colorbar('vert')
- colorbar('horiz')
- colorbar(h)
- h = colorbar(...)

描述：colorbar 函数在当前图形窗口中显示当前色图。

colorbar 更新大多数的色轴，或当当前轴无色轴时，colorbar 增加一个新的垂直色轴。

colorbar('vert')增加一个垂直色轴；colorbar('horiz')增加一个水平色轴。

colorbar(h)在 h 指定的位置放置一个色轴。如果该图形的宽度大于高度，作为色轴水平放置。

h = colorbar(...)返回一个指向色轴的句柄，属于轴的图形对象。

举例：

显示一个色轴：

```
surf(peaks(30))
```

```
colormap cool
colorbar
```

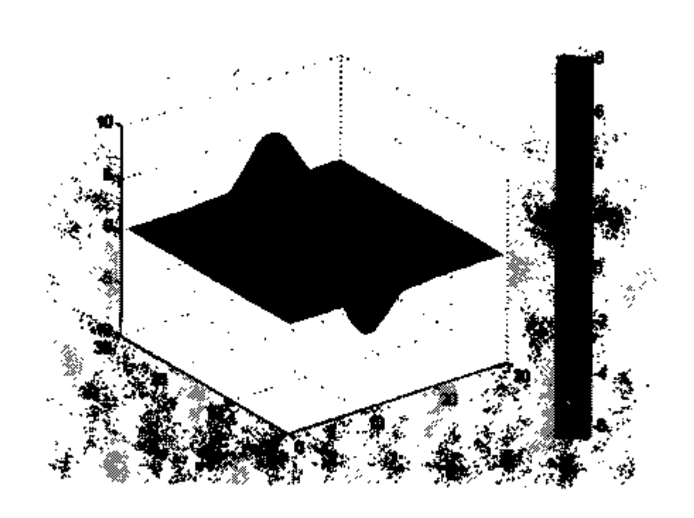


图 16-94 带色轴的表面图

4. 设置默认的属性值来显示不同的颜色方案

名称：colordef

设置默认的属性值来显示不同的颜色方案。

语法：该函数有如下五种表达形式：

- colordef white
- colordef black
- colordef none
- colordef(fig,color_option)
- h = colordef('new',color_option)

描述：colordef 使用户能够为图形的显示选择白色或黑色背景。它设置轴线和标签来显示面临的背景颜色。

colordef white 将轴的背景色设置为白色，将轴线和标签设置为黑色，将图形的背景色设置为淡灰色。

colordef black 将轴的背景色设置为黑色，将轴线和标签设置为白色，将图形背景色设置为暗灰色。

colordef none 将图形颜色设置为被 MATLAB4 使用过的颜色(特别是黑色背景)。

colordef(fig,color_option)设置由指向颜色选项'white'、'black'或'none'的句柄确定的图形窗口的颜色布局。

h = colordef('new',color_option)返回指向用指定的颜色选项创建的新图形的句柄(如，'white'，'black'或'none')。

5. 饱和色彩色图 HSV 向红绿蓝色图 RGB 转换

名称：hsv2rgb

饱和色彩色图 HSV 向红绿蓝色图 RGB 转换。

语法：M = hsv2rgb(H)

描述：M = hsv2rgb(H)转换饱和色彩色图 HSV 向红绿蓝色图 RGB。H 是一个 m×3 的矩阵，其中 m 是色图中颜色的数目。H 的列依次代表了色调，饱和度，强度。M 是一个 m×3 的矩阵，它的列分别代表了红绿蓝三色的强度。

6. 绘制色图

名称：rgbplot

绘制色图。

语法：rgbplot(cmap)

描述：rgbplot(cmap)用红，绿，蓝三色分别画出 cmap 矩阵的三个列向量，其中 cmap 是一个 m×3 的色图矩阵。

举例：

Plot the RGB values of the copper colormap.

rgbplot(copper)

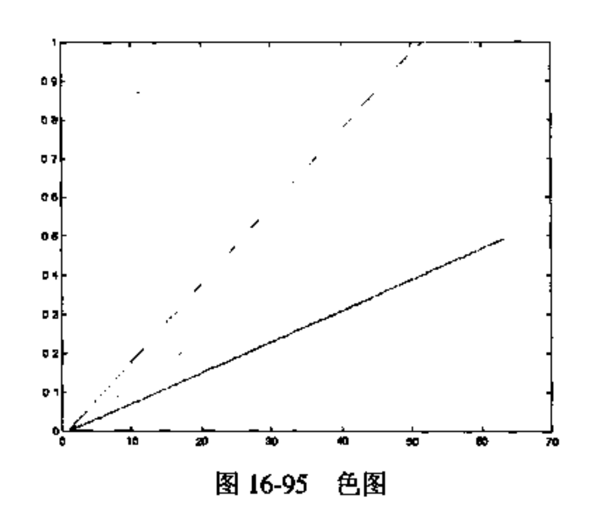


图 16-95　色图

7. 为灰度显示器设置默认的图形窗口属性

名称：graymon

为灰度显示器设置默认的图形窗口属性。

语法：graymon

描述：graymon 设置默认的图形属性，使灰度显示器有更清晰的显示。

8. 将 RGB 色图转换为 HSV 色图

名称：rgb2hsv

将 RGB 色图转换为 HSV 色图。

语法：cmap = rgb2hsv(M)

描述：cmap = rgb2hsv(M)将一个 RGB 色图 M 转化为一个 HSV 色图 cmap，两个色图都是 m×3 矩阵。两个色图的元素在 0 和 1 的范围内。

输入矩阵 M 的列，分别表示红、绿和蓝的亮度。输出矩阵的列 cmap 分别表示色调、饱和度和值。

9. 旋转色图

名称：spinmap

旋转色图。

语法：该函数有如下四种表达形式：

- spinmap
- spinmap(t)
- spinmap(t,inc)
- spinmap('inf')

描述：spinmap 函数按照某个增量循环变化色图 RGB 的值。如增量为 1，则颜色 1 变为颜色 2，颜色 2 变为颜色 3 等等。

spinmap 大约以 5 秒的周期按照增量为 2 循环变化色图。

spinmap(t)使色图的旋转周期为 10*t 秒。

spinmap(t,inc)使色图的旋转周期为 10*t 秒，变化的增量为 inc。

spinmap('inf')无限循环变化色图，直至按下 Ctrl-C。

10. 设置颜色渲染属性

名称：shading

设置颜色渲染属性。

语法：该函数有如下三种表达形式：

- shading flat
- shading faceted
- shading interp

描述：shading 函数控制着表面图形对象和块图形对象的颜色渲染。

shading flat 每一个网格线的片断和面都有一个恒定的颜色，它是由这个片断的终点或是具有最小索引的面的拐角颜色值确定的。

shading faceted 使用叠加的黑色网线来抹平渲染效果。这是默认的渲染模式。

shading interp 通过插值色图索引或沿着线或面的真色值，使得每一个线片和面都具有不同的颜色。

举例：

比较三种命令 flat、faceted 和 interp 下球面的渲染效果。

```
subplot(3,1,1)
sphere(16)
axis square
shading flat
title('Flat Shading')
subplot(3,1,2)
sphere(16)
axis square
shading faceted
title('Faceted Shading')
subplot(3,1,3)
sphere(16)
axis square
```

```
shading interp
title('Interpolated Shading')
```

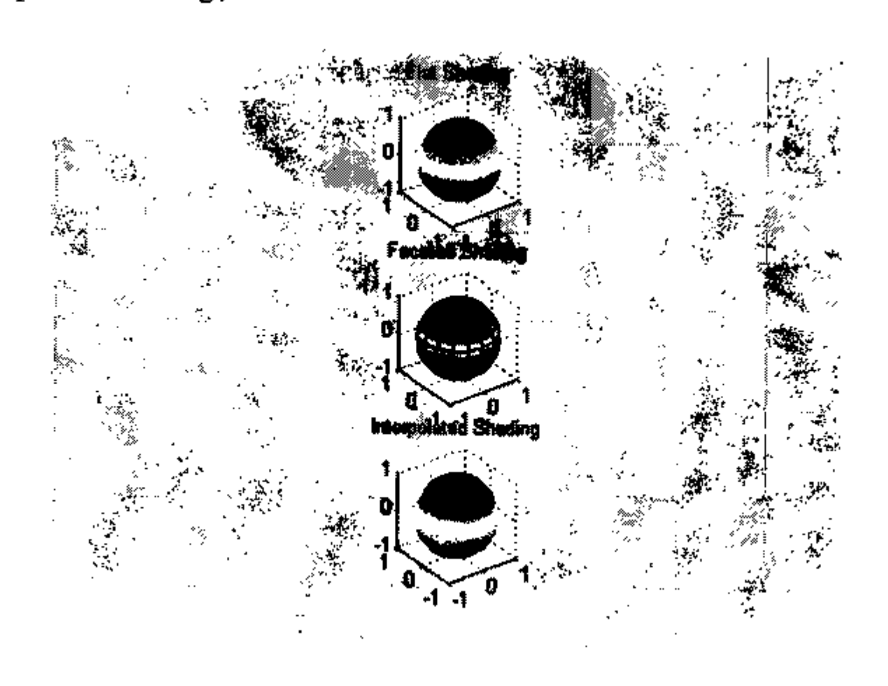


图 16-96　球面的渲染效果图

11. 计算和显示三维表面法向

名称：surfnorm

计算和显示三维表面法向。

语法：该函数有如下三种表达形式：

- surfnorm(Z)
- surfnorm(X,Y,Z)
- [Nx,Ny,Nz] = surfnorm(...)

描述：surfnorm 函数计算由 X、Y 和 Z 定义的表面法向。表面法向在每一个顶点都是非正则化的和有效的。在背离观察者的面元处，法向并不显示。

surfnorm(Z)和 surfnorm(X,Y,Z)画出一个表面及其表面法向。Z 是定义了表面 z 分量的一个矩阵。X 和 Y 是定义了表面 x 和 y 分量的向量或矩阵。

[Nx,Ny,Nz] = surfnorm(...)返回三维表面法向的分量。

举例：

画出截锥的法向向量。

```
[x,y,z] = cylinder(1:10);
surfnorm(x,y,z)
axis([-12 12 -12 12 -0.1 1])
```

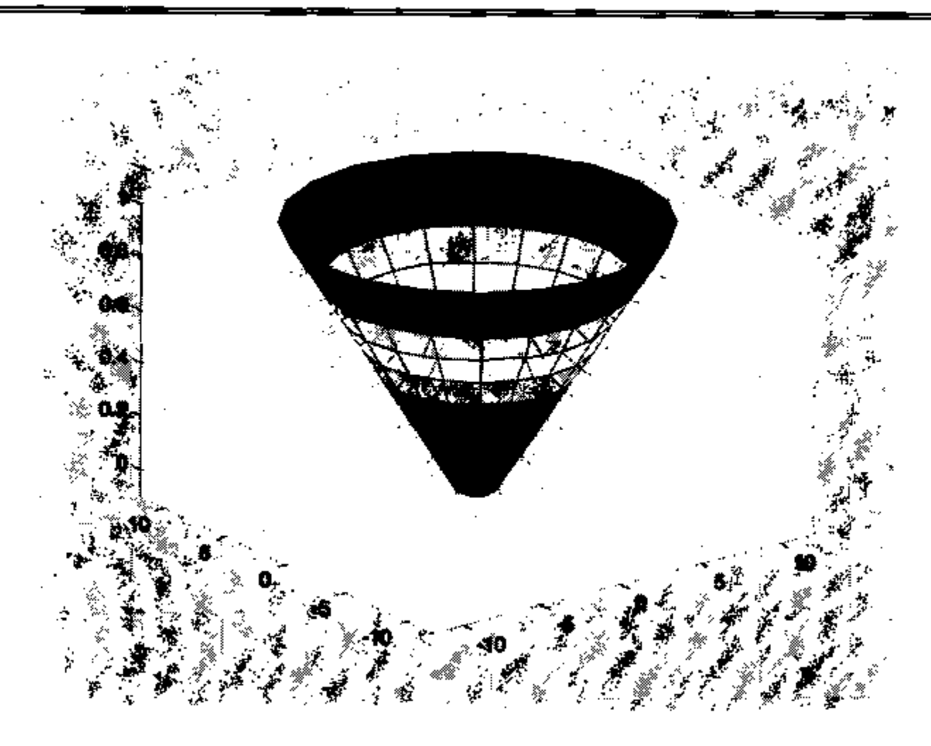


图 16-97 截锥的法向向量图

12. 改变轴的背景色

名称：whitebg

改变轴的背景色。

语法：该函数有如下四种表达形式：

- whitebg
- whitebg(h)
- whitebg(ColorSpec)
- whitebg(h,ColorSpec)

描述：whitebg 以当前图形窗口的背景颜色的补色作为背景色。

whitebg(h)对向量 h 指定的所有图形都以当前颜色的补色作为背景色。

whitebg(ColorSpec)和 whitebg(h,ColorSpec)按照 ColorSpec 的定义改变当前图形的轴的颜色。

举例：

设置背景色为蓝－灰色。

whitebg([0 .5 .6])

设置背景色为蓝色。

whitebg('blue')

13. 设置和获得当前色图

名称：colormap

设置和获得当前色图。

语法：该函数有如下三种表达形式：

- colormap(map)
- colormap('default')

- cmap = colormap

描述：色图是一个分量为 0.0 到 1.0 之间的 m×3 的实值矩阵。每一行是一个定义了一种颜色的 RGB 向量。色图的第 k 行定义了第 k 种颜色，这里 map(k,:) = [r(k) g(k) b(k)])指定了红色、绿色和蓝色的亮度。

colormap(map)将色图设置为矩阵映射。如果映射中的任何值在区间[0 1]之外，MATLAB 就返回一个错误信息，Colormap 的值必须在[0 1]之间。

colormap('default')将当前色图设置为默认色图。

cmap = colormap 检索当前色图。返回值在区间[0 1]之内。

颜色目录里的 M 文件产生大量的色图。每一个 M 文件承认色图的大小为一个选项。例如，colormap(hsv(128))就是用 128 种颜色创建一个 hsv 色图。如果用户不想确定一个尺寸，则 MATLAB 就创建一个大小与当前色图相同的色图。

MATLAB 支持大量的色图：

(1) autumn 从红色，通过橙色光滑变化到黄色。

(2) bone 是关于蓝色分量有较高值的灰度(grayscale)色图。这个色图可用于给 grayscale 图像方面增加"electronic"显示。

(3) colorcube 当试图提供灰色、纯红、纯绿和纯蓝之间的更多的间隔时，colorcube 包含了尽可能多的 RGB 里的颜色，这些颜色是按规则间隔排列的。

(4) cool 由暗的蓝绿色和洋红色组成。它从蓝绿色光滑变化到洋红。

(5) copper 从黑色光滑变化到亮铜色。

(6) flag 由红、白、蓝和黑色组成。这个色图就每一个索引增量来进行颜色的改变。

(7) gray 返回一个线性 grayscale 色图。

(8) hot 从黑色，通过暗红、橙色和黄色，光滑变化到白色。

(9) hsv 变化色彩－饱和度－值颜色模型的色彩分量。颜色开始于红色，经过黄色、绿色、蓝绿色、蓝色、洋红色，返回红色。色图对显示周期函数特别适当。hsv(m)与 hsv2rgb([h ones(m,2)])相同，这里 h 是线性的斜断面(ramp)，h = (0:m-1)'/m。

(10) jet 的范围是从蓝到红，并经过蓝绿色，黄色和橙色。它是 hsv 色图的一个变种。jet 色图与一个天体物理学的流动喷气模拟有关。

(11) lines 产生了由轴 ColorOrder 属性和暗灰色确定的颜色的色图。

(12) pink 包含柔暗的粉红色。pink 色图提供了灰度照片的棕褐色的调配。

(13) prism 重复红、橙、黄、绿、蓝和紫六种颜色。

(14) spring 由暗洋红和暗黄组成。

(15) summer 有暗绿和暗黄组成。

(16) white 是一个全白的单色色图。

(17) winter 由暗蓝和暗绿色组成。

举例：

(1) 用 jet 色图显示了流体喷气的数据。

```
load flujet
image(X)
colormap(jet)
```



图 16-98　气流图

(2)　显示人骨的 CAT 扫描图像。

```
load spine
image(X)
colormap bone
```

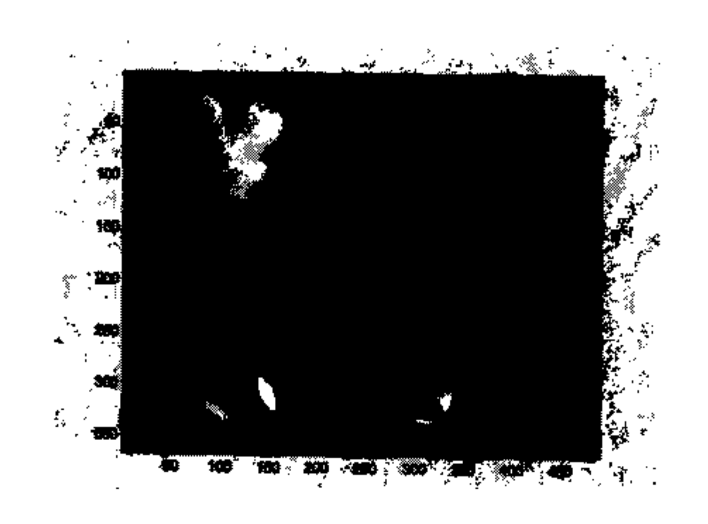


图 16-99　人骨的 CAT 扫描图

16.11　打印函数

1. 为打印输出设置纸张的方向

名称：orient

为打印输出设置纸张的方向。

语法：该函数有如下六种表达形式：

- orient
- orient landscape

- orient portrait
- orient tall
- orient(fig_handle), orient(simulink_model)
- orient(fig_handle,orientation), orient(simulink_model,orientation)

描述：orient 返回一个字符串表明当前纸张的方向，横向或纵向。

orient landscape 设置纸张方向为横向，即纸张最长的一边为水平方向，且图形居中，四周边距为 0.25inch。

orient portrait 设置纸张的方向为纵向，即纸张最长的一边为垂直方向，也是 MATLAB 的默认设置（注意用户修改图形属性时会影响 portrait 选项）。

orient tall 使得当前图形沿纵向铺满纸张的整个页面，只留下 0.25inch 的页边距。

orient(fig_handle)，orient(simulink_model)返回当前指定图形或 Simulink 模型的方向。

orient(fig_handle,orientation)，orient(simulink_model,orientation)为指定的图形或 Simulink 模型设置方向。

2. 创建硬拷贝输出

名称：print, printopt

创建硬拷贝输出。

语法：该函数有如下三种表达形式：

- print
- print -devicetype　-options filename
- [pcmd,dev] = printopt

描述：print 和 printopt 产生硬拷贝输出。关于 print 命令的所有选项都是可选择的。用户可以以它们的任何组合或任何顺序来使用它们。

print 将当前图形的内容，包括用户界面控制的位图表示，发送到使用由 printopt 定义的设备和系统打印命令的打印机上。

print –devicetype 指定了一个打印设备(一个输出格式，如 TIFF 或 PostScript 或控制 MATLAB 发送到打印机上内容的打印驱动器)，不考虑由 printopt 返回的值。设备区列出了所有支持的设备类型。

print –options 指定了修改打印命令方式的打印选项(例如，-noui 选项禁止用户界面控制的打印)。选项区列出了可用的选项。

print filename 将输出指向由 filename 指定的文件上。如果 filename 不包括扩展名，print 就附加上一个适当的扩展名，这依赖于设备(如.eps)。如果用户忽略文件名，print 将文件发送到默认的输出设备上(除了-dmeta 和-dbitmap，因为它们将输出放在剪贴板上)。

print(...)和圆括号里的选项一起构成打印的函数形式。可以使用户对任意输入项都能通过变量。这个形式对通过文件名和句柄很有用。

[pcmd,dev] = printopt 返回包含当前依赖于系统的打印命令和输出设备的字符串。printopt 是一个被打印到产生硬拷贝输出使用的 M 文件。用户能够编辑 M 文件 printopt.m 来设置用户的默认打印机类型和目的文件。

pcmd 和 dev 是依赖于平台的字符串。pcmd 包含用于发送文件到打印机的打印命令。dev 包含关于打印命令的设备选项。它们的默认设置是依赖于平台的。

表 16-3　　不同打印平台的 pcmd 和 dev 值

平台	pcmd	dev
UNIX	lpr -r -s	-dps2
VMS	PRINT/DELETE	-dps2
Windows	COPY /B %s LPT1:	-dwin

下表列出了由 MATLAB 内置驱动器支持的设备类型。一般地，在打印时，Level 2 PostScript 要比 Level 1 PostScript 文件小且快速。然而，并不是所有的 PostScript 打印机都支持 Level 2，因此在使用驱动器前要先确定打印机的兼容性。Level 2 PostScript 对 UNIX 和 VAX/VMS 是默认的。用户可以通过编辑 printopt.m 文件来改变默认值。

通过几乎所有的关于输入图像的字处理器，所有的平台均支持 TIFF 图像格式。JPEG 是一个有耗损的高压缩比的格式，在图像和 WWW 上的 HTML 文档的处理中，所有的平台都支持它。为了创建这些格式，MATLAB 使用 Z 缓冲器交付方法来交付图形，并且所得的像素映射被保存到指定的文件中。用户可以使用-r 分辨率开关来指定图像的分辨率。

表 16-4　　MATLAB 内置驱动器支持的设备类型表

设备	描述
-dps	Level 1 black and white PostScript
-dpsc	Level 1 color PostScript
-dps2	Level 2 black and white PostScript
-dpsc2	Level 2 color PostScript
-deps	Level 1 black and white Encapsulated PostScript (EPS)
-depsc	Level 1 color Encapsulated PostScript (EPS)
-deps2	Level 2 black and white Encapsulated PostScript (EPS)
-depsc2	Level 2 color Encapsulated PostScript (EPS)
-dhpgl	HPGL compatible with HP 7475A plotter
-dill	Adobe Illustrator 88 compatible illustration file
-dtiff	24-bit RGB TIFF with packbits compression (figures only)
-dtiffn	24-bit RGB TIFF with no compression (figures only)
-djpeg	Baseline JPEG image, quality factor defaults to 75 (figures only)
-djpegnn	Baseline JPEG image with nn (0-100) quality factor (figures only)

下表列出通过 Ghostscript 后处理器支持的附加设备，它将 PostScript 文件转换为其它格式。

表 16-5　　Ghostscript 后处理器支持的附加设备

设备	描述
-dlaserjet	HP LaserJet
-dljetplus	HP LaserJet+
-dljet2p	HP LaserJet IIP
-dljet3	HP LaserJet III
-ddeskjet	HP DeskJet and DeskJet Plus
-ddjet500	HP Deskjet 500
-dcdjmono	HP DeskJet 500C printing black only
-dcdjcolor	HP DeskJet 500C with 24 bit/pixel color and high-quality color (Floyd-Steinberg) dithering
-dcdj500	HP DeskJet 500C
-dcdj550	HP Deskjet 550C (UNIX only)
-dpaintjet	HP PaintJet color printer
-dpjxl	HP PaintJet XL color printer

续　表

设备	描述
-dpjetxl	HP PaintJet XL color printer
-dpjxl300	HP PaintJet XL300 color printer
-ddnj650c	HP DesignJet 650C
-dbj10e	Canon BubbleJet BJ10e
-dbj200	Canon BubbleJet BJ200
-dbjc600	Canon Color BubbleJet BJC-600 and BJC-4000
-dln03	DEC LN03 printer
-depson	Epson-compatible dot matrix printers (9- or 24-pin)
-depsonc	Epson LQ-2550 and Fujitsu 3400/2400/1200
-deps9high	Epson-compatible 9-pin, interleaved lines (triple resolution)
-dibmpro	IBM 9-pin Proprinter
-dbmp256	8-bit (256-color) BMP file format
-dbmp16m	24-bit BMP file format
-dpcxmono	Monochrome PCX file format
-dpcx16	Older color PCX file format (EGA/VGA, 16-color)
-dpcx256	Newer color PCX file format (256-color)
-dpcx24b	24-bit color PCX file format, three 8-bit planes
-dpbm	Portable Bitmap (plain format)
-dpbmraw	Portable Bitmap (raw format)
-dpgm	Portable Graymap (plain format)
-dpgmraw	Portable Graymap (raw format)
-dppm	Portable Pixmap (plain format)
-dppmraw	Portable Pixmap (raw format)

表 16-6　　用于 Windows 系统的附加设备

设备	描述
-dwin	使用 Windows 打印服务(黑和白)
-dwinc	使用 Windows 打印服务(彩色)
-dmeta	以增强 Windows 图元文件格式(彩色)
-dbitmap	以 Windows 位图格式复制到剪贴板(彩色)
-dsetup	显示打印启动对话框
-v	显示打印对象框的详细模式(默认是取消显示)

表 16-7　　用户指定的打印选项

值	作用
-tiff	将颜色 TIFF 的预览加入到密封的 PostScript
-loose	对 EPS 和 PS 设备使用稀疏耦合框
-cmyk	在 PostScript 里而不是在 RGB 里使用 CMYK 颜色
-append	追加到没有改写的已存在的 PostScript 文件
-rnumber	指定分辨率 Specify resolution in dots per inch
-adobecset	使用 PostScript 默认标志设置解码
-Pprinter	指定所用的打印机(仅限于 UNIX)
-fhandle	要打印的图形窗口对象的句柄(默认是当前图形窗口，见 gcf)
-swindowtitle	要打印的 SIMULINK 系统窗口的名字(默认是当前系统，见 gcs)
-painters	交付正在使用的打印机算法
-zbuffer	交付正在使用的 Z-buffer
-noui	禁止用户界面控制的打印

MATLAB 支持大量的标准纸张。用户可以通过设置了图形的 PaperType 属性列表来选择，或者从打印对话框中选择支持的纸型。

表 16-8　　图形的 PaperType 属性列表

属性值	尺寸(宽×高)
usletter	8.5×11 英寸
uslegal	11×14 英寸
tabloid	11×17 英寸
A0	841×1189 mm
A1	594×841 mm
A2	420×594 mm
A3	297×420 mm
A4	210×297 mm
A5	148×210 mm
B0	1029×1456 mm
B1	728×1029 mm
B2	514×728 mm
B3	364×514 mm
B4	257×364 mm
B5	182×257 mm
arch-A	9×12 英寸
arch-B	12×18 英寸
arch-C	18×24 英寸
arch-D	24×36 英寸
arch-E	36×48 英寸
A	8.5×11 英寸
B	11×17 英寸
C	17×22 英寸
D	22×34 英寸
E	34×43 英寸

3. 使用指定的格式保存图形或模型

名称：saveas

使用指定的格式保存图形或模型。

语法：该函数有如下两种表达形式：

- saveas(h,'filename.ext')
- saveas(h,'filename','format')

描述：saveas(h,'filename.ext')将带有句柄 h 的图形或模型保存到文件 filename.ext 中去。这个文件的格式由扩展名 ext 决定。ext 的允许值列在下表。

表 16-9　　ext 的允许值表

ext 值	格式
ai	Adobe Illustrator `88
bmp	Windows 位图
emf	Enhanced 图元文件
eps	EPS Level 1
fig	MATLAB 图形 (MATLAB 模型无效)
jpg	JPEG 图像 (MATLAB 模型无效)

续　表

ext 值	格式
m	MATLAB M-文件 (MATLAB 模型无效)
pbm	笔记本电脑位图
pcx	画笔 24-bit
pgm	笔记本灰图
png	笔记本网络图形
ppm	笔记本像素图
tif	TIFF 图像(被压缩过的)

saveas(h,'filename','format')将带有句柄 h 的图形或模型保存到使用指定格式的称为 filename 的文件中去。filename 可以有一个扩展名，但是扩展名并不用于定义文件格式。如果没有指定扩展名，对应于指定格式的标准扩展名自动加在文件上。

关于格式的允许值就是上表中的扩展名和由打印支持的设备类型。打印设备类型包括上面扩展名表中列出的格式和附加的文件格式。从上表中或从由打印支持的设备类型列表中使用扩展名。当用户对指定格式使用打印设备类型来另存时，不使用预先考虑的-d。

举例：

(1) 指定文件扩展名。

将当前图形保存为以.fig 为扩展名的文件 pred_prey.fig。用户可以用 Plot Editor 打开并进行编辑。

saveas(gcf,'pred_prey.fig')

(2) 指定文件格式，但没有扩展名。

使用 Adobe Illustrator 格式保存当前图形为文件 logo。从上表中使用.ai 的扩展名来指定文件格式。所创建的文件为 logo.ai。

saveas(gcf,'logo', 'ai')

(3) 指定文件格式和扩展名。

使用 Level 2 Color PostScript 格式将当前图形保存为文件 star.eps。如果用户使用 doc print 或 help print，用户可以从打印设备类型表里看到，这种格式的设备类型是-dpsc2，所创建的文件是 star.eps。

saveas(gcf,'star.eps', 'psc2')

16.12　图像处理（通用）

1. 复制图形对象及其子辈对象

名称：copyobj

复制图形对象及其子辈对象。

语法：new_handle = copyobj(h,p)

描述：copyobj 生成图形对象的拷贝。该拷贝除了其父辈和句柄不同外与原对象没有区别。拷贝的新的父辈必须兼容被拷贝的对象(例如，用户可以复制一个线对象到另一个轴对象)。

new_handle = copyobj(h,p)复制一个或多个由 h 所指向的对象，同时返回新对象的句柄

或句柄向量。新图形对象是 p 所指向的图形对象的子辈。

举例：

复制一个曲面图到另一个不同的图形窗口。

```
h = surf(peaks);
colormap hot
figure          % Create a new figure
axes            % Create an axes object in the figure
new_handle = copyobj(h,gca);
colormap hot
view(3)
grid on
```

注意当复制曲面时，色图（图形的属性），观测角度和网格（轴的属性）并不同时被复制。

2. 查找图形对象

名称：findobj

查找图形对象。

语法：该函数有如下四种表达形式：

- h = findobj
- h = findobj('PropertyName',PropertyValue,...)
- h = findobj(objhandles,...)
- h = findobj(objhandles,'flat','PropertyName',PropertyValue,...)

描述：findobj 定位图形对象并返回它们的句柄。用户可以使用特殊的属性值并沿指定的继承分支来限制搜索对象。

h = findobj 返回根对象及其所有后代的句柄。

h = findobj('PropertyName',PropertyValue,...) 返回所有将属性 PropertyName 设置为 PropertyValue 的图形对象句柄。用户可以指定多于一个的属性/值对，在这种情况下，findobj 只返回那些有指定值的对象。

h = findobj(objhandles,...)限制搜索列在 objhandles 及其后代中的对象

h = findobj(objhandles,'flat','PropertyName',PropertyValue,...)限制搜索那些列在 objhandles 里的对象，不搜索其后代。

举例：

在当前轴里查找所有的线对象。

```
h = findobj(gca,'Type','line')
```

3. 返回当前指向正在被调用的对象的句柄

名称：gcbo

返回当前指向正在被调用的对象的句柄。

语法：该函数有如下两种表达形式：

- h = gcbo
- [h, figure] = gcbo

描述：h = gcbo 返回一个指向图形对象的句柄，该图形对象正在被调用。

[h, figure] = gcbo 返回一个当前正在被调用的对象的句柄和包含此对象的图形窗口对象句柄。

4. 返回当前对象的句柄

名称：gco

返回当前对象的句柄。

语法：该函数有如下两种表达形式：

- h = gco
- h = gco(figure_handle)

描述：h = gco 返回当前对象的句柄。

h = gco(figure_handle)为由 figure_handle 指定的图形返回当前对象的值。

举例：

这个语句返回指向图形窗口 2 中的当前对象的句柄。

h = gco(2)

5. 获取对象的属性

名称：get

获取对象的属性。

语法：该函数有如下八种表达形式：

- get(h)
- get(h,'PropertyName')
- <m-by-n value cell array> = get(H,<property cell array>)
- a = get(h)
- a = get(0,'Factory')
- a = get(0,'FactoryObjectTypePropertyName')
- a = get(h,'Default')
- a = get(h,'DefaultObjectTypePropertyName')

描述：get(h)返回句柄 h 所指向的图形对象的所有属性及其取值。

get(h,'PropertyName')返回句柄 h 指向的图形对象的'PropertyName'属性的值。

<m-by-n value cell array> = get(H,pn)返回 m 个图形对象的 n 个值：到 m×n 的数组中，其中 m = length(H)，n 等于 pn 中所包含的属性名称的数目。

a = get(h)返回一个结构，其中的域名是对象的属性名称，相应的域值即属性的当前值。h 必须是标量。如果用户未指定输出参数，MATLAB 直接把信息显示在屏幕。

a = get(0,'Factory')返回所有用户能设置的属性的厂家设置值。a 是一个结构数组，其中的域名是对象的属性名称，相应的域值即属性的当前值。h 必须是标量。如果用户未指定输出参数，MATLAB 直接把信息显示在屏幕。

a = get(0,'FactoryObjectTypePropertyName')返回指定对象类型的属性的厂家设置值。参数 FactoryObjectTypePropertyName 是一个字，由 Factory 和对象类型（如,Figrue）及属性值（如，Color）相联结而成，即 FactoryFigureColor。

a = get(h,'Default')返回对象 h 当前所有的默认值。a 是一个结构数组，其中的域名是对

象的属性名称，相应的域值即属性的当前值。h 必须是标量。如果用户未指定输出参数，MATLAB 直接把信息显示在屏幕。

a = get(h,'DefaultObjectTypePropertyName')返回指定对象类型的属性的厂家设置值。参数 FactoryObjectTypePropertyName 是一个单词，由 Default 和对象类型（如,Figrue）及属性值（如，Color）相联结而成，即 DefaultFigureColor。

举例：

可以通过下列语句得到线型图形对象的 LineWidth 的默认值：

```
get(0,'DefaultLineLineWidth')
ans =
      0.5000
```

为了查询所有轴子对象的属性的值，可以定义一个属性名称的数组：

```
props = {'HandleVisibility', 'Interruptible';
         'SelectionHighlight', 'Type'};
output = get(get(gca,'Children'),props);
```

输出的变量放在一个数组中：

length(get(gca,'Children')-by-4.

例如：For example, type

```
patch;surface;text;line
output = get(get(gca,'Children'),props)
output =
     'on'    'on'    'on'    'line'
     'on'    'off'   'on'    'text'
     'on'    'on'    'on'    'surface'
     'on'    'on'    'on'    'patch'
```

6. 关于指定的方向旋转对象

名称：rotate

关于指定的方向旋转对象。

语法：该函数有如下两种表达形式：

- rotate(h,direction,alpha)
- rotate(...,origin)

描述：rotate 函数根据右手规则，在三维空间里旋转一个图形对象。

rotate(h,direction,alpha)将图形对象 h 旋转 alpha 度。direction 是描述与原点一起的旋转轴的一个二元素或三元素向量。

rotate(...,origin)指定旋转轴的原点为一个三维向量。默认原点是图形框的中心。

举例：

(1) 将一个图形对象绕 x 轴旋转 180 度。

```
h = surf(peaks(20));
rotate(h,[1 0 0],180)
```

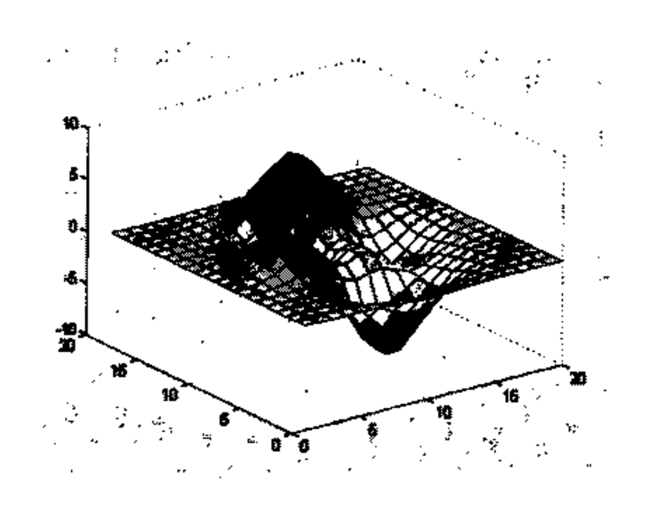


图 16-100　绕 x 轴选择 180⁰ 的图形

(2) 将一个表面图形对象沿 z 方向绕其中心旋转 45 度。

```
h = surf(peaks(20));
zdir = [0 0 1];
center = [10 10 0];
rotate(h,zdir,45,center)
```

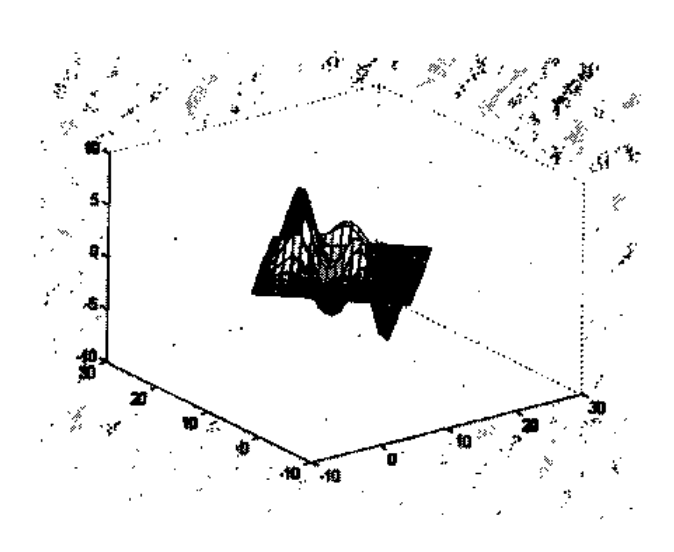


图 16-101　旋转表面图

7. 判断图形对象的句柄是否有效

名称：ishandle

判断图形对象的句柄是否有效。

语法：array = ishandle(h)

描述：array = ishandle(h)返回一个数组，其中元素为 1 表示所对应的句柄 h 中的元素是有效的；反之，等于 0 的值对应 h 中的元素是无效的。

举例：

判断先前由 fill 函数所返回的图形对象的句柄是否有效。

```
X = rand(4); Y = rand(4);
```

```
h = fill(X,Y,'blue')
delete(h(3))
ishandle(h)
ans =
     1
     1
     0
     1
```

8. 设置对象属性

名称：set

设置对象属性。

语法：该函数有如下十种表达形式：

- set(H,'PropertyName',PropertyValue,...)
- set(H,a)
- set(H,pn,pv...)
- set(H,pn,<m-by-n cell array>)
- a= set(h)
- a= set(0,'Factory')
- a= set(0,'FactoryObjectTypePropertyName')
- a= set(h,'Default')
- a= set(h,'DefaultObjectTypePropertyName')
- <cell array> = set(h,'PropertyName')

描述：set(H,'PropertyName',PropertyValue,...)在由 H 确定的对象上将已命名的属性设置为指定的值。H 可以是一个句柄向量，在这种情况下，对所有对象设置属性值。

set(H,a)在由 H 确定的对象上将已命名的属性设置为指定的值。a 是一个结构数组，其域名是对象属性名，其域值是对应的属性的值。

set(H,pn,pv,...)对所有在 H 里确定的对象，将在单元数组 pn 里确定的已命名的属性设置为单元数组 py 里的对应值。

set(H,pn,<m-by-n cell array>)在 m 个图形对象的每一个对象上设置 n 个属性值，这里 m = length(H)，n 等于包含在单元数组 pn 里的属性名的个数。它允许用户在每一个对象上将给定的一组属性设置为不同的值。

a = set(h)返回用户设置的属性和由 h 确定的对象的可能值。a 是一个结构数组，其域名是对象属性名，其域值值对应的属性的可能值。如果用户不指定一个输出项，MATLAB 就在屏幕上显示信息。h 必须是标量。

a = set(0,'Factory')返回这些属性，其默认值为对所有对象都是用户可设置的，并为每一个属性列出可能的值。a 是一个结构数组，其域名是对象属性名，其域值是对应属性的可能值。如果用户不指定一个输出项，MATLAB 就在屏幕上显示信息。

a = set(0,'FactoryObjectTypePropertyName')如果值是字符串，此命令就为指定的对象类型返回已命名的属性的可能值。选项 FactoryObjectTypePropertyName 是与对象类型(如 axes)和

属性名(如 CameraPosition)连接的字 Factory。

a = set(h,'Default')返回属性名，这个属性的默认值是设置到由 h 确定的对象上的。如果它们是字符串，set 也返回可能值。h 必须是标量。

a = set(h,'DefaultObjectTypePropertyName')为指定的对象类型返回已命名属性的可能值，如果这些值是字符串的话。选项 DefaultObjectTypePropertyName 是与对象类型(如 axes)和属性名(如 CameraPosition)连接的字 Default。例如 DefaultAxesCameraPosition.。h 必须是标量。

pv = set(h,'PropertyName')为已命名属性返回可能的值。如果这些可能值是字符串，set 返回单元数组 py 中单元里的每一个。对其它属性，set 返回一个空的单元数组。如果用户没有指定一个输出选项，MATLAB 就在屏幕上显示信息。h 必须是标量。

举例：

(1) 将当前轴的颜色属性设置为蓝色：

```
set(gca,'Color','b')
```

(2) 将图形里所有的线变为黑色：

```
plot(peaks)
set(findobj('Type','line'),'Color','k')
```

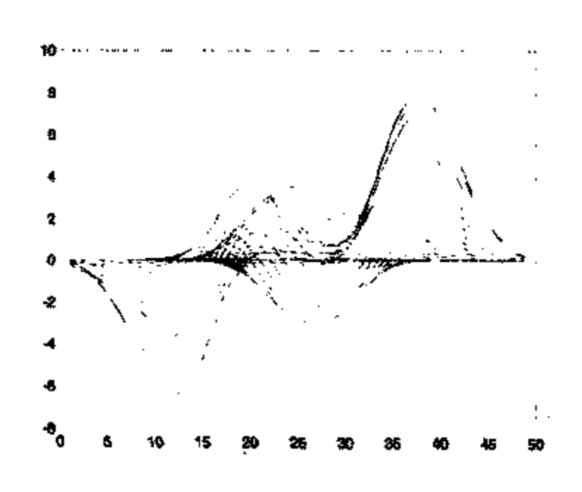


图 16-102 设置颜色的图形

用户可以在一个结构里定义一组属性来更好的组织用户的编码。例如，下面这些语句定义了一个名叫 active 的结构，它包含了一组在特殊的图形中为用户使用界面对象所用的属性定义。当这个图形变为当前图形时，MATLAB 改变颜色并激活控制功能。

```
active.BackgroundColor = [.7 .7 .7];
active.Enable = 'on';
active.ForegroundColor = [0 0 0];
if gcf == control_fig_handle
    set(findobj(control_fig_handle,'Type','uicontrol'),active)
end
```

用户可以使用单元数组，在每一个对象上，将属性设置为不同的值。例如，下面的语句为三个属性定义了一个单元数组：

```
PropName(1) = {'BackgroundColor'};
PropName(2) = {'Enable'};
PropName(3) = {'ForegroundColor'}
```

下面的语句定义了一个单元数组，对每一个对象，它都包含三个值(例如，一个 3×3 的单元单元)：

```
PropVal(1,1) = {[.5 .5 .5]};
PropVal(1,2) = {'off'};
PropVal(1,3) = {[.9 .9 .9]};
PropVal(2,1) = {[1 0 0]};
PropVal(2,2) = {'on'};
PropVal(2,3) = {[1 1 1]};
PropVal(3,1) = {[.7 .7 .7]};
PropVal(3,2) = {'on'};
PropVal(3,3) = {[0 0 0]}
```

现在通过选项来进行设置：

```
set(H,PropName,PropVal)
```

这里 length(H) = 3，并且每一个单元都是指向一个用户界面控制的句柄。

16.13 图像处理（对象创建）

1. 生成轴图形对象

名称：axes

生成轴图形对象。

语法：该函数有如下四种表达形式：

- axes
- axes('PropertyName',PropertyValue,...)
- axes(h)
- h = axes(...)

描述：axes 是生成轴类图形对象的低层函数。

axes 在当前图形中生成一个轴对象，其属性为默认值。

axes('PropertyName',PropertyValue,...)生成具有指定值的一个轴对象。除了用户所明确指定的外，MATLAB 采用默认值。

axes(h)令已经存在的轴 h 为当前轴。并且该轴对象在当前图形的子辈对象中为第一根轴，把图形对象的 CurrentAxes 的值作为 h 的值。而当前轴可以作为绘制图像，线，块，表面和文本等对象的目标函数。

h = axes(...)返回指向所生成的轴对象的句柄。

举例：

(1) 放大。

使用纵横比和刻度范围放大：

```
sphere
set(gca,'DataAspectRatio',[1 1 1],...
        'PlotBoxAspectRatio',[1 1 1],'ZLim',[-0.6 0.6])
```

利用照相机的视角放大：

```
sphere
set(gca,'CameraViewAngle',get(gca,'CameraViewAngle')-5)
set(gca,'CameraViewAngle',get(gca,'CameraViewAngle')+5)
```

(2) 决定轴的位置。

用户可以通过设置轴的位置值来确定轴在图形窗口的位置。如：

h = axes('Position',position_rectangle)在当前图形窗口中生成一个轴对象，并返回该对象的句柄。其中轴的大小和位置由一个四元素向量所定义的矩形所确定，

position_rectangle = [left, bottom, width, height];

上面向量中的 left 和 bottom 项定义了从图形窗口的左下角到矩形的左下角的距离。width 和 height 项定义了矩形的宽度和高度。上述数值的单位由 Units 属性确定。默认时，MATLAB 使用规格化单位，即(0, 0)是图形窗口的左下角的坐标，(1.0,1.0)是右上角的坐标。

用户可以在一个图形窗口中定义多个轴：

```
axes('position',[.1  .1  .8  .6])
mesh(peaks(20));
axes('position',[.1  .7  .8  .2])
pcolor([1:10;1:10]);
```

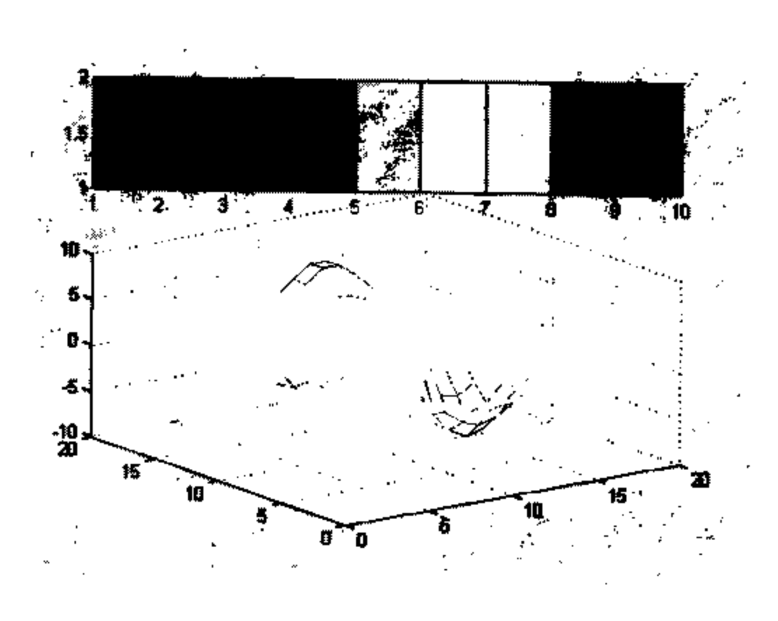


图 16-103　定义色轴的图形

在此例中，第一个图占据了图形的下面三分之二的部分，第二个图占据了上面的三分之一部分。

对象继承表：

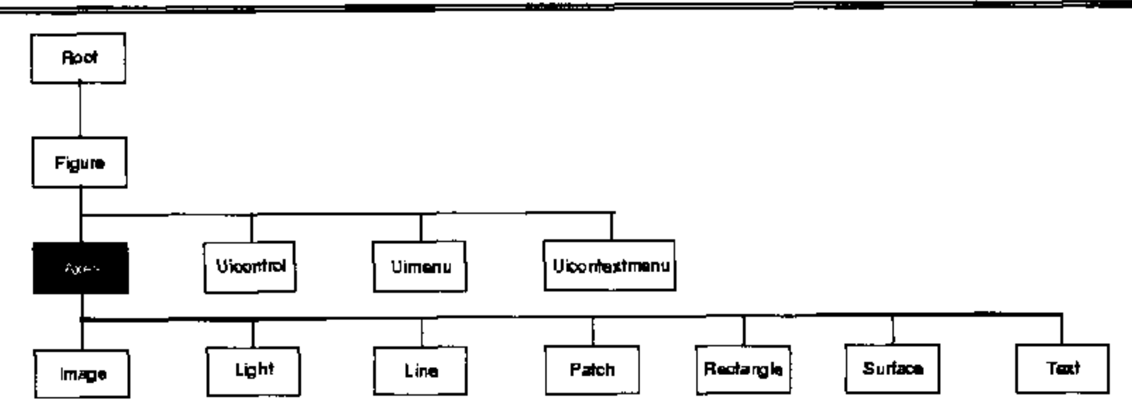


图 16-104 对象继承表

设置默认值：

用户可以在图形和根的层次上设置轴的默认值：

set(0,'DefaultAxesPropertyName',PropertyValue,...)

set(gcf,'DefaultAxesPropertyName',PropertyValue,...)

其中 PropertyName 是轴的名称属性，PropertyValue 便是所设置的数值。使用 set 和 get 函数可设置和获取轴值：。

下面的表格显示了轴对象的所有属性及其描述。

表 16-10 轴对象的属性列表

属性名称	属性描述	属性值
控制样式和外观		
Box	坐标轴是否成封闭状	值：on, off；默认值：off
Clipping	对轴对象无效	
GridLineStyle	网格线的线型	值：-, --, :, -., none； 默认值：: (dotted line)
Layer	坐标轴在图形的层面	值：bottom, top；默认值：bottom
LineStyleOrder	多条线对象的对应线型的序列	值：LineSpec；默认值：-
LineWidth	线宽	值：点数；默认值：0.5 倍的点
SelectionHighlight	高亮度显示被选中的轴	值：on, off；默认值：on
TickDir	刻度向内向外	值：in, out；默认值：in (2-D), out (3-D)
TickDirMode	刻度方向的模式	值：auto, manual；默认值：auto
TickLength	轴上刻度记号的长度	值：[2-D 3-D]；默认值：[0.01 0.025]
Visible	轴可见或不可见	值：on, off；默认值：on
XGrid, YGrid, ZGrid	是否画相应轴方向上的网格线	值：on, off ；默认值：off
轴的通用信息		
Children	图形上面各对象（images, lights, lines, patches, surfaces,和 text）的句柄	值：句柄向量
CurrentPoint	最后一次鼠标单击的位置	值：2×3 的矩阵
HitTest	控制被单击轴是否成为当前对象	值：on, off；默认值：on
Parent	图形窗口的句柄	值：图形句柄标量
Position	轴在图形窗口的位置和大小	值：[left bottom width height]； 默认值：[0.1300 0.1100 0.7750 0.8150]
Selected	是否显示轴被选中	值：on, off；默认值：on
Tag	用户指定的标注	值：任意字符串；默认值：''(空串)

续　表

属性名称	属性描述	属性值
Type	图形对象的类型（只读）	值：字符串：'axes'
Units	描述 position 属性数值的单位	值：　inches, centimeters, characters, normalized, points, pixels； 默认值：normalized
UserData	用户指定的数据	值：任意矩阵；默认值：[] (空矩阵)
选择字体和标注		
FontAngle	字体是否倾斜	值：normal, italic, oblique； 默认值：normal
FontName	字体名称	值：用户系统中所安装的字体； 默认值：Helvetica 字体
FontSize	字体大小	值：一个整数；默认值：10
FontUnits	描述字体大小所用的单位	值：points, normalized, inches, centimeters, pixels；默认值：points
FontWeight	字体是否加粗	值：normal, bold, light, demi； 默认值：normal
Title	标题	值：任意合法的文本对象的句柄
XLabel, YLabel, ZLabel	轴的标注	值：任意合法的文本对象的句柄
XTickLabel, YTickLabel, ZTickLabel	刻度记号的标注	值：字符串矩阵； 默认值：MATLAB 自动选择的数值矩阵
XTickLabelMode, YTickLabelMode, ZTickLabelMode	刻度记号的标注的模式	值：auto, manual； 默认值：auto
控制轴的缩放比例		
XAxisLocation	x 轴的位置	值：top, bottom；默认值：bottom
YAxisLocation	y 轴的位置	值：right left；默认值：left
XDir, YDir, ZDir	各轴数值增加的方向	值：normal, reverse；默认值：normal
XLim, YLim, ZLim	各轴的数值范围	值：[min max]；默认值：MATLAB 自动选择的最小值 min 和最大值 max
XLimMode, YLimMode, ZLimMode	各轴的数值范围的控制模式	值：auto, manual；默认值：auto
XScale, YScale, ZScale	各轴的刻度方式	值：线性，对数；默认值：线性
XTick, YTick, ZTick	各轴刻度标记的位置	值：数值向量；默认值：MATLAB 自动确定的位置
XTickMode, YTickMode, ZTickMode	各轴刻度标记位置的控制方式	值：auto, manual；默认值：auto
控制视角		
CameraPosition	观测点的位置	值：[x,y,z]轴坐标；默认值：有 MATLAB 自动确定
CameraPositionMode	观测点位置的控制方式	值：auto, manual；默认值：auto
CameraTarget	被观测的物体的中心点	值：[x,y,z] 轴坐标； 默认值：有 MATLAB 自动确定
CameraTargetMode	被观测的物体的中心点的控制方式	值：auto, manual；默认值：auto
CameraUpVector	观测方向	值：[x,y,z] 轴坐标； 默认值：有 MATLAB 自动确定
CameraUpVectorMode	观测方向的控制方式	值：auto, manual；默认值：auto
CameraViewAngle	观测角的范围	值：0 度到 180 度的任意角度； 默认值：有 MATLAB 自动确定
CameraViewAngleMode	观测角范围的控制方式	值：auto, manual； 默认值：auto

续 表

属性名称	属性描述	属性值
Projection	投影方式	值：orthographic, perspective； 默认值：orthographic
控制轴的纵横比		
DataAspectRatio	数据纵横比	值：三个比值[dx dy dz]； 默认值：有 MATLAB 自动确定
DataAspectRatioMode	数据纵横比的控制方式	值：auto, manual；默认值：auto
PlotBoxAspectRatio	绘图框的纵横比	值：三个比值[dx dy dz]； 默认值：有 MATLAB 自动确定
PlotBoxAspectRatioMode	绘图框的纵横比的控制方式	值：auto, manual；默认值：auto
控制调用函数执行		
BusyAction	调用函数时遇到中断如何处理	值：cancel, queue；默认值：queue
ButtonDownFcn	按钮按下时执行的调用函数	值：字符串；默认值：空字符串
CreateFcn	创建轴时执行的调用函数	值：字符串；默认值：空字符串
DeleteFcn	删除轴时执行的调用函数	值：字符串；默认值：空字符串
Interruptible	执行调用函数时是否接收中断	值：on, off；默认值：on
UIContextMenu	轴所相关的上下文菜单	值：上下文菜单的句柄
指定交付模式		
DrawMode	绘图方式	值：normal, fast；默认值：normal
为图形显示指定目标轴		
HandleVisibility	控制是否能获取轴的句柄	值：on, callback, off；默认值：on
NextPlot	下一个图形的绘制方式	值：add, replace, replacechildren； 默认值：replace
定义颜色的属性		
AmbientLightColor	屏幕的背景色	值：ColorSpec；默认值：[1 1 1]
CLim	数据与色图的映射	值：[cmin cmax]； 默认值：有 MATLAB 自动确定
CLimMode	数据与色图映射的方式	值：auto, manual；默认值：auto
Color	轴的背景色	值：none, ColorSpec；默认值：none
ColorOrder	线的颜色	值：m×3 的 RGB 矩阵； 默认值：确定于使用的颜色设置
XColor, YColor, ZColor	轴和刻度的颜色	值：ColorSpec； 默认值：确定于当前的颜色设置

2. 创建一个图形窗口

名称：figure

创建一个图形窗口。

语法：该函数有如下四种表达形式：

- figure
- figure('PropertyName',PropertyValue,...)
- figure(h)
- h = figure(...)

描述：figure 创建图形窗口对象。图形窗口对象是屏幕上的一个单个的窗口，在这里 MATLAB 可以显示图形输出。

figure 使用默认属性值创建一个新的图形窗口对象。

figure('PropertyName',PropertyValue,...)使用指定的属性值创建一个新的图形窗口对象。MATLAB 对用户没有明确定义为选项的属性使用默认的值。

figure(h)要做一件事，它依赖于是否存在一个带句柄 h 的图形窗口。如果 h 是指向一个已存在的图形窗口的句柄，figure(h)使由 h 确定的图形窗口成为当前的图形窗口，使其可视化并将其放置在屏幕上所有其它的图形窗口之上。当前图形窗口是关于图形输出的目标。如果 h 不是指向一个已经存在的图形窗口的句柄，但是一个整数，figure(h)创建一个图形窗口，并且给它分配一个句柄 h。如果 h 不是指向一个图形窗口的句柄，也不是一个整数，figure(h)就发生了错误。

h = figure(...)返回指向图形窗口对象的句柄。

举例：

创建一个有屏幕的四分之一大小的图形窗口，它位于屏幕的左上方。

scrsz = get(0,'ScreenSize');

figure('Position',[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2])

对象继承表：

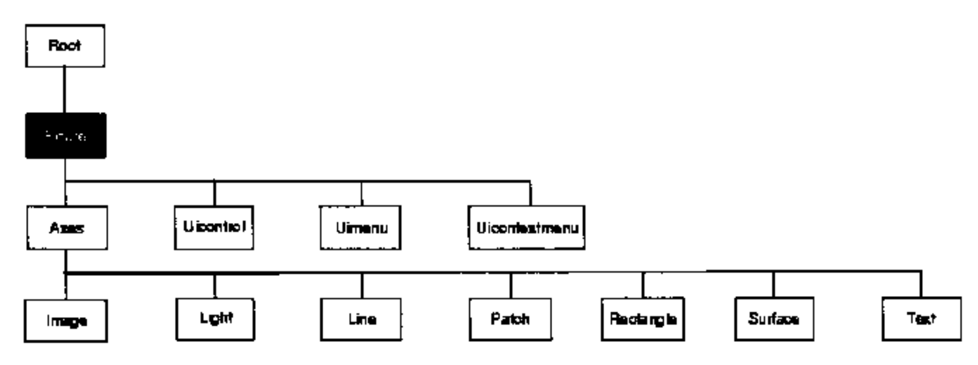


图 16-105　对象继承表

用户可以只在根层设置图形窗口属性：set(0,'DefaultFigureProperty',PropertyValue...)，这里 Property 是图形属性的名字，PropertyValue 是用户指定的值。使用 set 和 get 来访问图形窗口属性。

表 16-11　　图形窗口对象的属性列表

属性名称	属性描述	属性值
图形窗口的定位		
Position	图形窗口的位置和大小	值：一个四元素向量[left, bottom, width, height]；默认值：依赖于显示
Units	用于解释 Position 属性	值：英寸、厘米、规格化、点、像素、字符；默认值：像素
指定样式和外观		
Color	图形背景颜色	值：ColorSpec；默认值：依赖于颜色方案(见 colordef)
MenuBar	切换图形窗口菜单条	值：none，figure；默认值：figure
Name	图形窗口标题	值：字符串；默认值：''(空字符)
Numbertitle	显示"Figure No. n"，这里 n 是图形窗口数	值：on，off；默认值：on
Resize	确定是否使用鼠标调整图形窗口的尺寸	值：on，off；默认值：on
SelectionHighlight	突出图形窗口	值：on，off；默认值：on
Visible	决定图形窗口是否可视	值：on，off；默认值：on
WindowStyle	现在 normal 或 model 窗口	值：Values: normal, modal；默认值：normal

续 表

属性名称	属性描述	属性值
控制色图		
Colormap	图形窗口色图	值：RGB 值的 m×3 矩阵；默认值：jet 色图
Dithermap	在伪色显示器上的用于真色数据的色图	值：RGB 值的 m×3 矩阵； 默认值：具有全部色彩的色图
DithermapMode	启动 MATLAB 生成的颤动图	值：自动，手动；默认值：手动
FixedColors	Colors not obtained from colormap	值：RGB 值的 m×3 矩阵(只读)
MinColormap	Minimum number of system color table entries to use	值：标量；默认值：64
ShareColors	允许 MATLAB 共享系统颜色表存取窗口	值：on、off；默认值：on
确定交付者		
BackingStore	关闭屏幕像素缓存	值：on、off；默认值：on
DoubleBuffer	为简单的动画进行自由闪烁交付	值：on、off；默认值：on
Renderer	用于屏幕和打印的交付方法	值：指针、z 缓冲器、OpenGL； 默认值：由 MATLAB 自动选择 on
图形窗口的通用信息		
Children	图形上面各对象（uicontrol、uimenu 和 uicontextmenu）的句柄	值：句柄向量
Parent	图形窗口的句柄	值：图形句柄标量
Selected	是否显示轴被选中	值：on、off；默认值：on
Tag	用户指定的标注	值：任意字符串；默认值：''(空串)
Type	图形对象的类型（只读）	值：'axes'字符串
UserData	用户指定的数据	值：任意矩阵；默认值：[] (空矩阵)
RendererMode	自动或用户选择的交付者	值：auto、manual；默认值：auto
当前状态的信息		
CurrentAxes	图形窗口里当前轴的句柄	值：轴句柄
CurrentCharacter	图形窗口里最后按下的键	值：单字符(只读)
CurrentObject	图形窗口里当前对象的句柄	值：图形对象句柄
CurrentPoint	在图形窗口里击下的最后按钮的位置	值：两元素向量[x-coord, y-coord]
SelectionType	鼠标选择类型(只读)	值：normal, extended, alt, open
调用函数执行		
BusyAction	处理调用函数时遇到的中断	值：cancel, queue；默认值：queue
ButtonDownFcn	按钮按下时执行的调用函数	值：字符串；默认值：空字符串
CloseRequestFcn	调用 close 命令时定义一个执行的调用函数	值：字符串；默认值：空字符串
CreateFcn	创建图形窗口时执行的调用函数	值：字符串；默认值：空字符串
DeleteFcn	删除图形时执行的调用函数(通过 close 和 delete)	值：字符串；默认值：空字符串
Interruptible	执行调用函数时是否接收中断	值：on, off；默认值：on(可被中断)
KeyPressFcn	在图形窗口里按下一个键时定义一个执行的调用函数	值：字符串；默认值：空字符串
ResizeFcn	当图形窗口的尺寸改变时，定义一个执行的调用函数	值：字符串；默认值：空字符串

续　表

属性名称	属性描述	属性值
WindowButtonDownFcn	在图形窗口里按下鼠标时定义一个执行的调用函数	值：字符串；默认值：空字符串
WindowButtonMotionFcn	在图形窗口里移动鼠标时定义一个执行的调用函数	值：字符串；默认值：空字符串
WindowButtonUpFcn	在图形窗口里释放鼠标时定义一个执行的调用函数	值：字符串；默认值：空字符串
控制访问对象		
IntegerHandle	指定整数或非整数图形窗口句柄	值：on、off；默认值：on(整数句柄)
HandleVisibility	确定是否用于句柄的可视化	值：on、callback、off；默认值：on
HitTest	确定一个图形窗口是否变为当前对象	值：on、off；默认值：on
NextPlot	确定如何向图形窗口显示附加的图形	值：add、replace、replacechildren； 默认值：add
定义指针		
Pointer	选择指针符号	值：crosshair, arrow, watch, topl, topr, botl, botr, circle, cross, fleur, left, right, top, bottom, fullcrosshair, ibeam, custom； 默认值：arrow
PointerShapeCData	定义指针的数据	值：16×16 矩阵；默认值：将指针设置为 costum
PointerShapeHotShop	确定指针活动地点	值：两元素向量[row, column]；默认值：[1,1]
影响打印的属性		
InvertHardcopy	为打印改变图形颜色	值：on，off；默认值：on
PaperOrientation	纸张的水平和垂直定位	值：portrait，landscape；默认值：portrain
PaperPosition	在打印纸张上控制图形窗口的位置	值：四元素向量[left, bottom, width, height]
PaperPositionMode	使 WYSIWYG 能够打印图形	值：auto，manual；默认值：auto
PaperSize	在 PaperUnits 指定的当前纸张类型的尺寸	值：[width, height]
PaperType	从标准纸张尺寸中选择	值：见属性描述；默认值：usletter
PaperUnits	用于确定 PaperSize 和 PaperPosition 的 Units	值：normalized, inches, centimeters, pointsinches； 默认值：inches
控制 XWindows 显示(只限于 Unix)		
XDisplay	为 MATLAB 指定显示	值：显示标识符；默认值：0.0
XVisual	选择用于 MATLAB 的 visual	值：visual ID
XVisualMode	visual 的自动或手动选择	值：auto, manual；默认值：auto

3. 显示图像对象

名称：image

显示图像对象。

语法：该函数有如下五种表达形式：

- image(C)
- image(x,y,C)
- image(...,'PropertyName',PropertyValue,...)
- image('PropertyName',PropertyValue,...) Formal synatx - PN/PV only
- handle = image(...)

描述：image 通过矩阵元素直接映射到色图上而生成一个图像对象。

image 函数有两种形式：一种是高层函数，调用 newplot 函数，确定在什么位置绘制图像，并且设置相应轴对象的属性：XLim 和 YLim 属性可以用框线封闭图像；Layer 属性值设为 top 可以把图形放置在刻度和网格线的上层面；YDir 可设置沿反方向增加；View 可设置为[0 90]。另一种是低层函数，不用调用 newplot，直接在当前窗口中增添绘制图像。低层函数的参数列表只能包括属性名称及值对。

用户可以设置属性名称及属性值对为数据对，结构数组和空数组。

image(C)把矩阵 C 中的元素以图像形式显示出来。C 中的每一个元素定义了图像中的一块矩形片断的颜色。

image(x,y,C)中的 x 和 y 是二维向量，定义了 x 和 y 轴的范围，不过生成的图像和 image(C)的结果相同。如果用户想使轴的标注和物理现实的尺寸大小相当，这个函数会有帮助。

image(x,y,C,'PropertyName',PropertyValue,...)是定义了属性名称和数值对的一个高层函数，在绘制图像前需要调用 newplot 函数。

image('PropertyName',PropertyValue,...)是一个底层函数，输入参数只有属性名称和数值对。

handle = image(...)返回指向所生成的图像对象的句柄。

对象继承表：

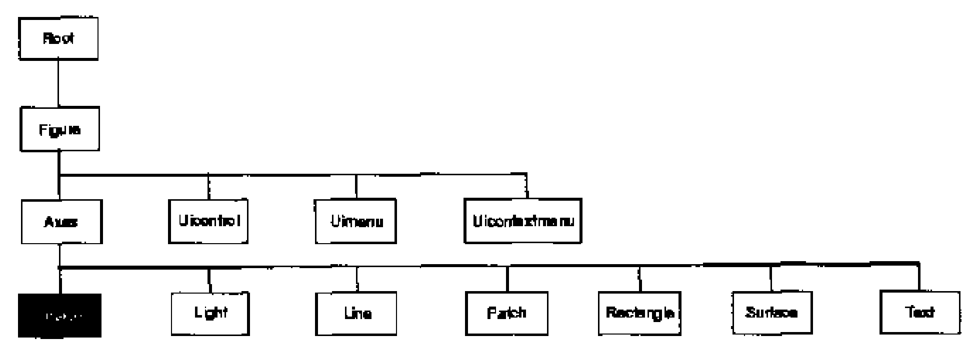


图 16-106　对象继承表

用户能在轴，图形和根的层次上设置图形属性的默认值：

set(0,'DefaultImageProperty',PropertyValue...)

set(gcf,'DefaultImageProperty',PropertyValue...)

set(gca,'DefaultImageProperty',PropertyValue...)

其中 PropertyName 是图像的属性名，PropertyValue 便是所设置的数值。使用 set 和 get 函数可设置和获取轴值。

表 16-12　　图像对象的属性列表

属性名称	属性描述	属性值
生成图像的数据		
CData	图像数据	值：矩阵或 m×3 数组； 默认值：enter image;axis image ij
CDataMapping	数据到色图的映射	值：scaled, direct；默认值：direct
XData	图像在 x 轴方向上的位置	值：[min max]； 默认值：[1 size(CData,2)]
YData	图像在 y 轴方向上的位置	值：[min max]； 默认值：[1 size(CData,1)]

续　表

属性名称	属性描述	属性值
控制外观		
Clipping	剪贴	值：on, off；默认值：on
EraseMode	绘制和擦除图像的方式	值：normal,none,xor,background；默认值：normal
SelectionHighlight	图像被选中时高亮度显示	值：on, off；默认值：on
Visible	图像是否可见	值：on, off；默认值：on
控制对象		
HandleVisibility	确定图像对象的句柄是否和什么时候是可见的	值：on, callback, off；默认值：on
HitTest	确定图像是否为当前对象	值：on, off；默认值：on
图像的通用信息		
Children	无子对象	值：[] (空矩阵)
Parent	父对象通常是 axes 轴对象	值：　轴句柄
Selected	是否显示图像被选中	值：on, off；默认值：on
Tag	用户指定的标注	值：任意字符串；默认值：'' (空串)
Type	图像对象的类型（只读）	值：字符串：'axes'
UserData	用户指定的数据	值：任意矩阵；默认值：[] (空矩阵)
控制调用函数执行		
BusyAction	调用函数时遇到中断如何处理	值：cancel, queue；默认值：queue
ButtonDownFcn	按钮按下时执行的调用函数	值：字符串；默认值：空字符串
CreateFcn	创建图像时执行的调用函数	值：字符串；默认值：空字符串
DeleteFcn	删除图像时执行的调用函数	值：字符串；默认值：空字符串
Interruptible	执行调用函数时是否接收中断	值：：on, off；默认值：on
UIContextMenu	图像所相关的上下文菜单	值：上下文菜单的句柄

4. 创建一个照明对象

名称：light

创建一个照明对象。

语法：该函数有如下两种表达形式：

- light('PropertyName',PropertyValue,...)
- handle = light(...)

描述：light 在当前轴里创建一个照明对象。它只影响块对象和面对象。

light('PropertyName',PropertyValue,...)使用对已命名的属性指定的值创建一个照明对象。除非用户指定了另外的带有父属性的轴，否则 MATLAB 就将当前轴指定为父照明。

handle = light(...)返回创建的照明对象的句柄。

举例：

用位于无穷远处，方向为[1 0 0](即沿 x 轴)的光源照明 peaks 表面。

```
h = surf(peaks);
set(h,'FaceLighting','phong','FaceColor','interp',...
      'AmbientStrength',0.5)
light('Position',[1 0 0],'Style','infinite')
```

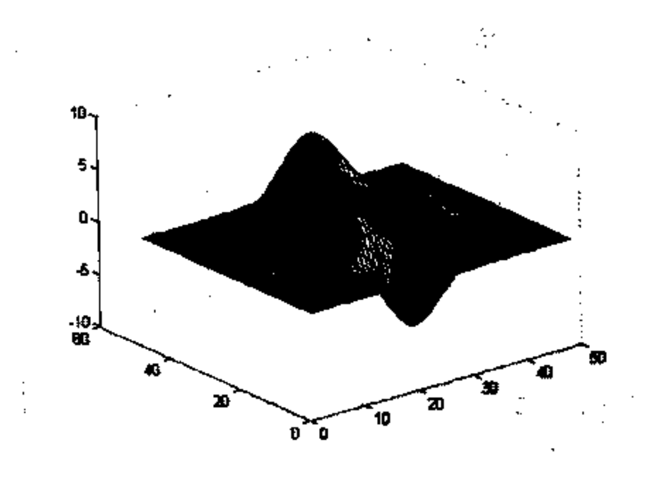


图 16-107　照明图

对象继承表：

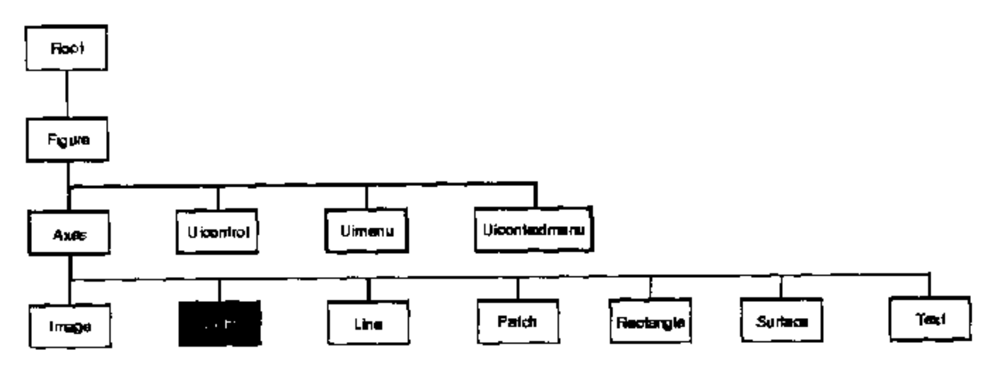


图 16-108　对象继承表

用户可以在轴上、图形窗口上和根层上设置照明属性：

set(0,'DefaultLightProperty',PropertyValue...)

set(gcf,'DefaultLightProperty',PropertyValue...)

set(gca,'DefaultLightProperty',PropertyValue...)

这里 Property 是照明属性的名字，并且 PropertyValue 是用户指定的值。使用 set 和 get 命令可以访问照明属性

表 16-13　　照明属性列表

属性名称	属性描述	属性值
定义照明		
Color	由照明对象产生的照明的颜色	值：ColorSpec
Position	轴上照明的定位	值：轴单元里的 x、y、z 坐标系；默认值：[1 0 1]
Style	平行或分叉光源	值：infinite, local
控制外观		
SelectionHighlight	不被照明对象使用	值：on、off；默认值：on
Visible	使照明效果可视或不可视	值：on、off；默认值：on
控制对象的访问		
HandleVisibility	确定线句柄是否对其它函数可视或什么时候可视	值：on, callback, off；默认：on

续　表

属性名称	属性描述	属性值:
HitTest	不被照明对象使用	值：on、off；默认：on
照明的一般信息		
Children	没有子照明	值：[] (空矩阵)
Parent	照明对象的父对象永远是轴对象	值：轴句柄
Selected	不被照明对象使用	值：on、off；默认：on
Tag	用户指定的标注	值：任何字符串； 默认：'' (空字符串)
Type	图形对象的类型(只读)	值：字符串'light'
UserData	用户指定的数据	值：任何矩阵； 默认：[] (空矩阵)
有关调用函数执行的属性		
BusyAction	调用函数时遇到中断如何处理	值：cancel, queue； 默认值：queue
ButtonDownFcn	按钮按下时执行的调用函数	值：字符串； 默认值：空字符串
CreateFcn	创建照明时执行的调用函数	值：字符串； 默认值：空字符串
DeleteFcn	删除照明时执行的调用函数	值：字符串； 默认值：空字符串
Interruptible	执行调用函数时是否接收中断	值：on, off；默认值：on
UIContextMenu	照明所相关的上下文菜单	值：上下文菜单的句柄

5. 生成线对象

名称：line

生成线对象。

语法：该函数有如下五种表达式：

- line(X,Y)
- line(X,Y,Z)
- line(X,Y,Z,'PropertyName',PropertyValue,...)
- line('PropertyName',PropertyValue,...) low-level-PN/PV pairs only
- h = line(...)

描述：line 在当前图形中生成一个线对象。用户可指定线的颜色，宽度，线型和标记和其它特征。

line 函数有两种形式：

(1)当用户使用非正式的语法格式指定坐标数据时，line 会选择自动颜色和线型的样式(即前三个参数看作数据坐标)，line(X,Y,Z)，MATLAB 类似 plot 函数，自动在轴的 ColorOrder 和 LineStyleOrder 值：中循环选择颜色和线型，区别在于 line 不调用 newplot 函数；

(2)纯低层函数。当用户调用有属性名和值的 line 时，如 line('XData',x,'YData',y,'ZData',z)，MATLAB 在当前窗口中按默认值绘制线对象。注意此时用户不能指定坐标矩阵。

line(X,Y)在当前窗口上按照向量 X 和 Y 的定义增加线对象。如果 X 和 Y 是同阶的矩阵，line 函数对矩阵的每一列数据生成一个线对象。

line(X,Y,Z)在三维坐标系中绘制线对象。

line(X,Y,Z,'PropertyName',PropertyValue,...)按照给出的属性名和值对和其它属性的默认

值生成一个线对象。

line('XData',x,'YData',y,'ZData',z,'PropertyName',PropertyValue,...)按照输入的属性参数值在当前窗口中生成一个线对象。如同上面的其它非正式的语法形式，这个低层的函数形式不接收数据坐标矩阵。

h = line(...)返回一个指向 line 函数所生成的每一个线对象的句柄列向量。

举例：

此例使用 line 函数给一个现成的曲线加阴影。首先绘制该曲线，并保存该曲线的句柄：

```
t = 0:pi/20:2*pi;
hline1 = plot(t,sin(t),'k');
```

然后，通过偏移 x 坐标起到阴影的作用，并且阴影线为淡灰色，宽度略宽于默认值：

```
hline2 = line(t+.06,sin(t),'LineWidth',4,'Color',[.8 .8 .8]);
```

最后，令第一条曲线显示在前面：

```
set(gca,'Children',[hline1 hline2])
```

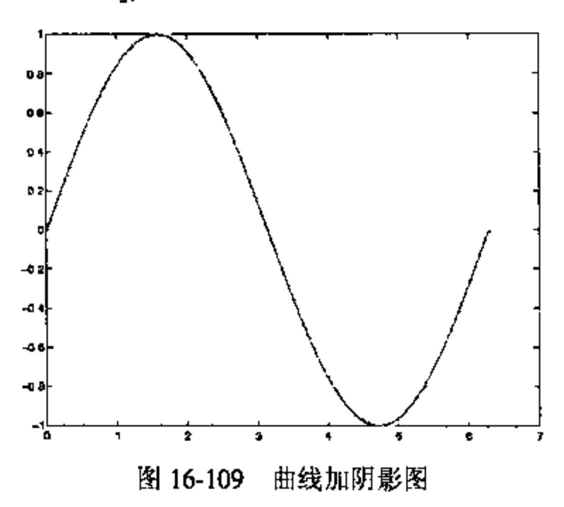


图 16-109　曲线加阴影图

对象继承表：

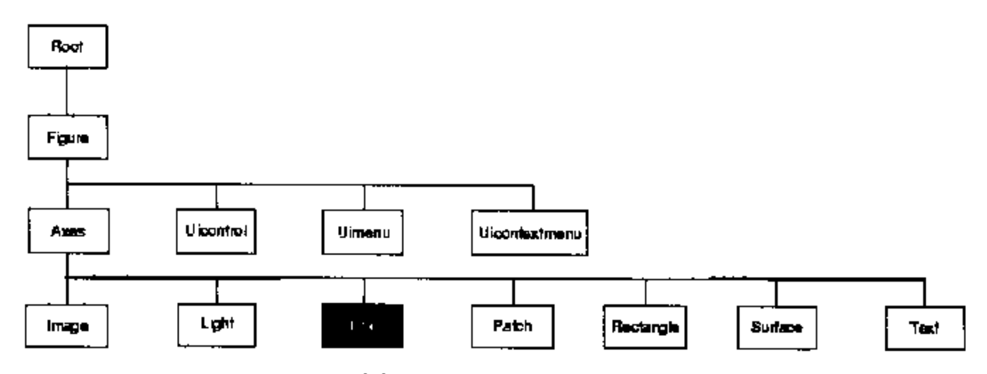


图 16-110　对象继承表

用户能在轴，图形和根的层次上设置图形属性的默认值：

```
set(0,'DefaultLinePropertyName',PropertyValue,...)
set(gcf,'DefaultLinePropertyName',PropertyValue,...)
set(gca,'DefaultLinePropertyName',PropertyValue,...)
```

其中 PropertyName 是线的属性名，PropertyValue 便是所设置的数值。使用 set 和 get 函数可设置和获取线的值。

表 16-14　　线对象的属性列表

属性名称	属性描述	属性值
生成线的数据		
XData	x 坐标值	值：向量或矩阵；默认值：[0 1]
YData	y 坐标值	值：向量或矩阵；默认值：[0 1]
ZData	z 坐标值	值：向量或矩阵； 默认值：[] empty matrix
控制线的样式和标记		
LineStyle	线型	值：-, --, :, -., none；默认值：-
LineWidth	线宽	值：标量；默认值：0.5 点
Marker	数据点的标记符号	值：参见 Marker property； 默认值：none
MarkerEdgeColor	数据标记符号边界的颜色	值：ColorSpec, none, auto； 默认值：auto
MarkerFaceColor	数据标记符号内部填充的颜色	值：ColorSpec, none, auto； 默认值：none
MarkerSize	数据标记符号的大小	值：用点数计算的尺寸； 默认值：6
控制外观		
Clipping	剪贴	值：on, off；默认值：on
EraseMode	绘制或擦除线的方式	值：normal, none, xor, background；默认值：normal
SelectionHighlight	高亮度显示被选中的线	值：on, off；默认值：on
Visible	线可见或不可见	值：on, off；默认值：on
Color	线的颜色	ColorSpec
控制对象		
HandleVisibility	控制是否能获取轴的句柄	值：on, callback, off； 默认值：on
HitTest	控制被单击轴是否成为当前对象	值：on, off；默认值：on
线的一般信息		
Children	线对象无子对象	值：空矩阵
Parent	线对象的父对象通常是轴对象	值：轴句柄标量
Selected	是否显示线被选中	值：on, off；默认值：on
Tag	用户指定的标注	值：任意字符串； 默认值：''(空串)
Type	线对象的类型（只读）	值：字符串：'axes'
UserData	描述 position 属性数值的单位	值：inches, entimeters, characters, normalized, points, pixels； 默认值：normalized
控制调用函数时中断的执行		
BusyAction	调用函数时遇到中断如何处理	值：cancel, queue；默认值：queue
ButtonDownFcn	按钮按下时执行的调用函数	值：字符串；默认值：空字符串
CreateFcn	创建线时执行的调用函数	值：字符串；默认值：空字符串
DeleteFcn	删除线时执行的调用函数	值：字符串；默认值：空字符串
Interruptible	执行调用函数时是否接收中断	值：on, off；默认值：on
UIContextMenu	线所相关的上下文菜单	值：上下文菜单的句柄

6. 创建块图形对象

名称：patch

创建块图形对象。

语法：该函数有如下五种表达形式：

- patch(X,Y,C)
- patch(X,Y,Z,C)
- patch(...'PropertyName',PropertyValue...)
- patch('PropertyName',PropertyValue...) PN/PV pairs only
- handle = patch(...)

描述：patch 是关于创建块图形对象的低层图形函数。一个块对象是由其顶点坐标定义的一个或多个多边形。用户可以指定块的颜色和亮度。

patch(X,Y,C)向当前轴增加填充的二维块。X 和 Y 的元素确定了一个多边形的顶点。如果 X 和 Y 是矩阵，MATLAB 就每一列画出一个多边形。C 决定块的颜色。它可以是一个单个的 ColorSpec，每一面一个颜色或每一个顶点一个颜色。如果 C 是 1×3 向量，则假设它是直接指定了颜色的一个 RGB。patch(X,Y,Z,C)在三维坐标系下创建一个块。

patch(...'PropertyName',PropertyValue...)沿着带有 property name/property value 对的 X、Y、(Z)和 C 的选项来指定附加的块属性。patch('PropertyName',PropertyValue,...)使用 property name/property value 对来确定所有的属性。这个形式使用户忽略颜色说明，因为 MATLAB 使用默认的面颜色和边颜色，除非用户能将一个值分配给 FaceColor 和 EdgeColor 属性。这个形式也允许用户使用 Faces 和 Vertices 属性指定块，而不是使用 x、y 和 z 轴坐标系。

handle = patch(...)返回它创建的块对象的句柄。

下表概括了 MATLAB 如何解释由 CData 和 FaceVertexCData 属性定义的颜色数据 CData 属性的。

表 16-15　　CData 属性的数据

[X,Y,Z]数据	CData Required for		结果
维数	索引	真色	
m×n	标量	1×1×3	使用对所有块的面指定的单色。边只能使用单色。
m×n	1×by×n(n >=4)	1×n×3	对每一个块面使用一个颜色。边只能使用单色
m×n	m×n	m×n×3	将一个颜色分配给每一个顶点。块面可以是 flat(一种单色)或 interpolated，边可以是 flat 或 interpolated

表 16-16　　FaceVertexCData 属性的解释

顶点	面	FaceVertexCData Required for		结果
维数	维数	索引	真色	
m×n	k×3	标量	1×3	使用对所有块的面指定的单色。边只能使用单色。
m×n	k×3	k×1	k×3	对每一个块面使用一个颜色。边只能使用单色
m×n	k×3	m×1	m×3	将一个颜色分配给每一个顶点。块面可以是 flat (一种单色)或 interpolated，边可以是 flat 或 interpolated

举例：

(1)　使用指定 X、Y 和 Z 坐标系的方法创建块对象。

```
x = [0 0;0 1;1 1];
y = [1 1;2 2;2 1];
z = [1 1;1 1;1 1];
tcolor(1,1,1:3) = [1 1 1];
tcolor(1,2,1:3) = [.7 .7 .7];
patch(x,y,z,tcolor)
```

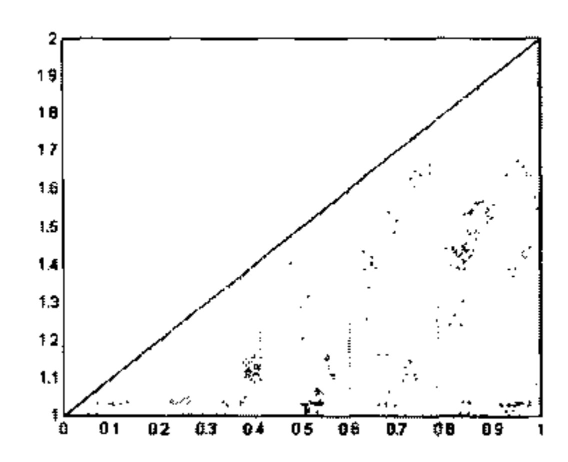


图 16-111　块对象图

(2)　使用指定 Vertices 和 Faces 的方法创建一个块对象

```
vert = [0 1 1;0 2 1;1 2 1;1 1 1];
fac = [1 2 3;1 3 4];
tcolor = [1 1 1;.7 .7 .7];
patch('faces',fac,'vertices',vert,'FaceVertexCData',tcolor,...
        'FaceColor','flat')
```

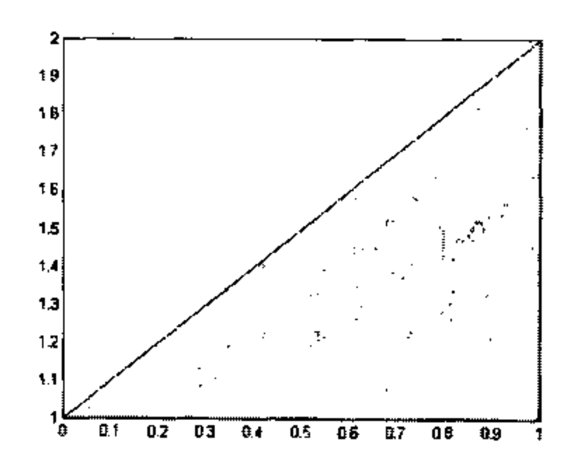


图 16-112　块对象图

对象继承表：

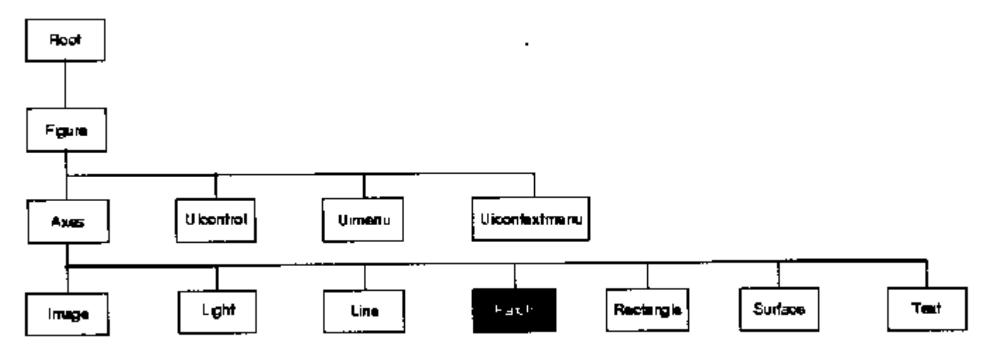


图 16-113　对象继承表

用户可以在轴上，图形窗口上和根层上设置块属性：

set(0,'DefaultPatchPropertyName',PropertyValue...)

set(gcf,'DefaultPatchPropertyName',PropertyValue...)

set(gca,'DefaultPatchPropertyName',PropertyValue...)

其中，PropertyName 是块属性的名字，并且 PropertyValue 是用户指定的值。使用 set 和 get 命令可以访问属性属性。

表 16-17　　**块对象属性列表**

属性名称	**属性描述**	**属性值**
定义对象的数据		
Faces	关于顶点的联系矩阵	值：m×n 矩阵；默认值：[1,2,3]
Vertices	顶点的 xyz 坐标系的矩阵(和 Faces 一起使用)	值：矩阵；默认值：[0,1;1,1;0,0]
XData	块的顶点的 x 坐标	值：向量或矩阵；默认值：[0;1;0]
YData	块的顶点的 y 坐标	值：向量或矩阵；默认值：[1;1;0]
ZData	块的顶点的 z 坐标	值：向量或矩阵； 默认值：[](空矩阵)
指定颜色		
CData	与 XData/YData/ZData 方法一起使用的颜色数据	值：标量、向量或矩阵； 默认值：[](空矩阵
CDataMapping	控制 CData 到色图的映射	值：scaled, direct； 默认值：scaled, direct
EdgeColor	面上边的颜色	值：ColorSpec, none, flat, interp； 默认值：ColorSpec
FaceColor	面的颜色	值：ColorSpec, none, flat, interp； 默认值：ColorSpec
FaceVertexCData	与 Faces/Vertices 方法一起使用的颜色数据	值：矩阵； 默认值：[](空矩阵
MarkerEdgeColor	标记的颜色或用于填充标记的边的颜色	值：ColorSpec, none, auto； 默认值：none
MarkerFaceColor	为有闭合形状的标记填色	值：ColorSpec, none, auto； 默认值：none
控制照明的效果		
AmbientStrength	周围光的强度	值：标量≥0 和≤1； 默认值：0.3
DiffuseStrength	弥散光的强度	值：标量≥0 和≤1；默认值：0.6

续　表

属性名称	属性描述	属性值
BackFaceLighting	控制指向背离照相机的面的照明	值：unlit, lit, reverselit； 默认值：reverselit
EdgeLighting	用于照明边的方法	值：none, flat, gouraud, phong； 默认值：none
FaceLighting	用于照明边的方法	值：none, flat, gouraud, phong； 默认值：none
NormalMode	MATLAB 生成的或用户指定的法向向量	值：auto, manual； 默认值：auto
SpecularColorReflectance	镜像反射光的混合色	值：标量 0 到 1；默认值：1
SpecularExponent	镜像反射的粗糙度	值：标量≥1；默认值：10
SpecularStrength	镜像光的强度	值：标量≥0 和≤1；默认值：0.6
VertexNormals	顶点法向向量	值：矩阵
定义边和标记		
LineStyle	从五种线样式中选择	值：-, --, :, -., none；默认值：-
LineWidth	用点数计的边的宽度	值：标量；默认值：0.5 点
Marker	在数据点绘出标记	值：见 Marker 属性；默认值：none
MarkerSize	用点数计算标记的尺寸	值：点数的大小；默认值：6
控制外观		
Clipping	剪贴	值：on, off；默认值：on
EraseMode	绘制或擦除线的方式	值：normal, none, xor, background；默认值：normal
SelectionHighlight	高亮度显示被选中的线	值：on, off；默认值：on
Visible	线可见或不可见	值：on, off；默认值：on
控制对象的访问		
HandleVisibility	决定块句柄是否对其它函数可视及什么时候可视	值：on, callback, off； 默认值：on
HitTest	决定块是否变为当前对象	值：on,off；默认值：on
Controlling Callback Routine Execution		
BusyAction	调用函数时遇到中断如何处理	值：cancel, queue； 默认值：queue
ButtonDownFcn	按钮按下时执行的调用函数	值：字符串； 默认值：空字符串
CreateFcn	创建块时执行的调用函数	值：字符串； 默认值：空字符串
DeleteFcn	删除块时执行的调用函数	值：字符串； 默认值：空字符串
Interruptible	执行调用函数时是否接收中断	值：on、off； 默认值：on
UIContextMenu	与块所相关的上下文菜单	值：上下文菜单的句柄
块的通用信息		
Children	没有子块的块对象	值：[] (空矩阵)
Parent	父块对象永远是轴对象	值：轴句柄
Selected	指明块是否处于"selected"状态	值：on、off；默认：on
Tag	用户指定的标注	值：任何字符串； 默认值：'' (空字符串)
Type	图形对象的类型(只读)	值：字符串'light'
UserData	用户指定的数据	值：任何矩阵； 默认值：[] (空矩阵)

7. 生成二维矩形对象

名称：rectangle

生成二维矩形对象。

语法：该函数有如下四种表达形式：

- rectangle
- rectangle('Position',[x,y,w,h])
- rectangle(...,'Curvature',[x,y])
- h = rectangle(...)

描述：rectangle 绘制一个矩形。

rectangle('Position',[x,y,w,h])绘制一个矩形，起始点在(x, y)，宽度是 w，高度是 h。注意：如果想以指定的比例显示矩形，用户必须设置图形的纵横比，才能保证一个单位长度正比于 x 和 y 轴的单位长度。

rectangle(...,'Curvature',[x,y])指定矩形的四边的曲率，使得矩形变成一个椭圆形。水平曲率 x 定义了矩形上下边的曲率；垂直曲率 y 定义了矩形的左右边的曲率。

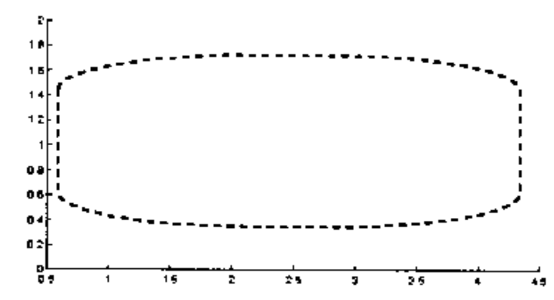
x 和 y 的取值范围从 0（无曲率）到 1（最大曲率）。曲率取值为[0,0]时生成一个矩形；曲率取值为[1,1]时生成一个椭圆形；如果只定义了一个曲率的值，则水平和垂直的边都取同一个曲率。弯曲的大小取决于短的一边。

h = rectangle(...)返回一个指向矩形对象的句柄。

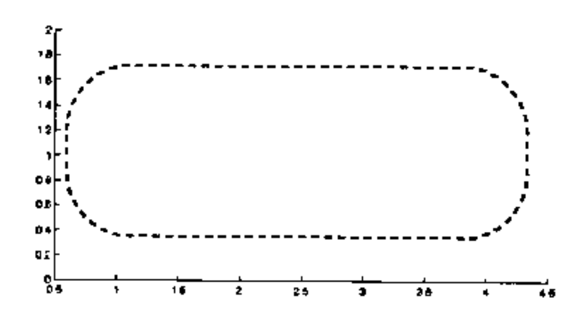
举例：

(1) 下例设置数据的纵横比为[1, 1, 1]，矩形便按照指定的比例显示。注意水平和垂直的曲率可以不等。同样，注意只给定一个曲率值的效果。

```
rectangle('Position',[0.59,0.35,3.75,1.37],...
          'Curvature',[0.8,0.4],...
         'LineWidth',2,'LineStyle','--')
daspect([1,1,1])
```



指定一个曲率值[0.4]：



曲率 [1] 生成一个短边为半圆的矩形：

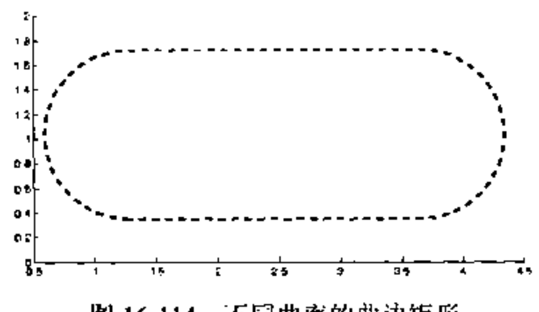


图 16-114　不同曲率的曲边矩形

(2) 下例生成一个红色的椭圆。

```
rectangle('Position',[1,2,10,5],'Curvature',[1,1],...
          'FaceColor','r')
daspect([1,1,1])
xlim([0,12])
ylim([1,8])
```

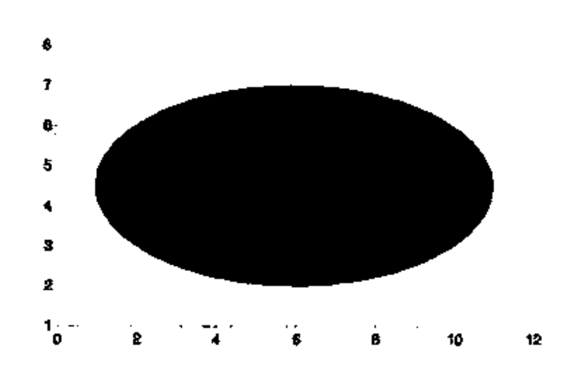


图 16-115　红色椭圆图

对象层次表：

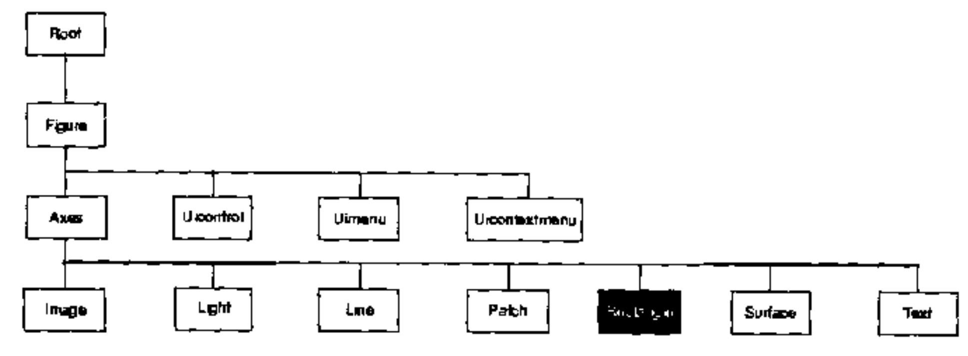


图 16-116　对象继承表

用户能在轴，图形和根的层次上设置图形属性的默认值：

```
set(0,'DefaultRectangleProperty',PropertyValue...)
```

set(gcf,'DefaultRectangleProperty',PropertyValue...)

set(gca,'DefaultRectangleProperty',PropertyValue...)

其中 PropertyName 是矩形的属性名，PropertyValue 便是所设置的数值。使用 set 和 get 函数可设置和获取矩形的值：。

下面的表格显示了矩形对象的所有属性及其描述。

表 16-18　　　　矩形对象属性列表

属性名称	属性描述	值：
定义矩形的对象		
Curvature	水平或垂直的曲率	值：元素值在 0 和 1 之间的一个二元素向量；默认值：[0,0]
EraseMode	绘制或擦除矩形的方式	值：normal, none, xor, background；默认值：normal
EdgeColor	矩形边界的颜色	值：Colorspec or none；默认值：ColorSpec [0,0,0]
FaceColor	矩形内部填充的颜色	值：Colorspec or none；默认值：none
LineStyle	边界线型	值：-, --, :, -., none；默认值：-
LineWidth	边界线宽	值：标量；默认值：点数为 0.5
Position	矩形的位置，高度和宽度	值：[x,y,width,height]；默认值：[0,0,1,1]
矩形的通用信息		
Children	无子对象	
Parent	父对象总是轴对象	值：轴句柄标量
Selected	是否显示矩形被选中	值：on, off；默认值：on
Tag	用户指定的标注	值：任意字符串；默认值：''(空串)
Type	图形对象的类型（只读）	值：字符串：'axes'
UserData	描述 position 属性数值的单位	值：inches, entimeters, characters, normalized, points, pixels；默认值：normalized
控制调用函数的执行		
BusyAction	调用函数时遇到中断如何处理	值：cancel, queue；默认值：queue
ButtonDownFcn	按钮按下时执行的调用函数	值：字符串；默认值：空字符串
CreateFcn	创建矩形时执行的调用函数	值：字符串；默认值：空字符串
DeleteFcn	删除矩形时执行的调用函数	值：字符串；默认值：空字符串
Interruptible	执行调用函数时是否接收中断	值：on, off；默认值：on
UIContextMenu	矩形所相关的上下文菜单	值：上下文菜单的句柄
控制对象		
HandleVisibility	控制是否能获取矩形的句柄	值：on, callback, off；默认值：on
HitTest	控制被单击矩形是否成为当前对象	值：on, off；默认值：on
控制外观		

续　表

属性名称	属性描述	值：
Clipping	剪贴	值：on, off；默认值：on
SelectionHighlight	高亮度显示被选中的矩形	属性值：on, off; 默认值：on
Visible	矩形可见或不可见	属性值：on, off 默认值：on

8. 创建面对象

名称：surface

创建面对象。

语法：该函数有如下六种表达形式：

- surface(Z)
- surface(Z,C)
- surface(X,Y,Z)
- surface(X,Y,Z,C)
- surface(...'PropertyName',PropertyValue,...)
- h = surface(...)

描述：surface 是用于创建面图形对象的低层次函数。surface 是使用每一个元素的行索引和列索引创建的矩阵数据的图形。

surface(Z)画出由矩阵 Z 确定的面。这里，Z 是一个单值函数，它定义在一个矩形网格上。

surface(Z,C)画出由 Z 和颜色指定的面。颜色是由 C 确定的。

surface(X,Y,Z)使用 C = Z，因此颜色是与 x-y 平面上的面高度比例的。

surface(X,Y,Z,C)画出由 X、Y 和 Z 确定的参数面，颜色由 C 指定。

surface(x,y,Z), surface(x,y,Z,C)用向量替换头两个矩阵项，它们必须有 length(x) = n 和 length(y) = m，这里[m,n] = size(Z)。在这种情况下，面的顶点是(x(j),y(i),Z(i,j))。注意 x 对应于 Z 的列，y 对应于 Z 的行。

surface(...'PropertyName',PropertyValue,...)跟随带有 property name/property value 对的 X、Y、Z 和 C 选项指定附加的面属性。

h = surface(...)返回指向已创建的表面对象的一个句柄。

举例：

本例使用 peaks M 文件生成的数据创建一个面，并使用小丑(clown)图像来进行着色。ZData 是一个 49×49 的矩阵，CData 是一个 200×320 的矩阵。为了能够使用不同阶数的 ZData 和 CData，用户必须将面的 FaceColor 设置为 testuremap。

```
load clown
surface(peaks,flipud(X),...
        'FaceColor','texturemap',...
        'EdgeColor','none',...
        'CDataMapping','direct')
colormap(map)
view(-35,45)
```

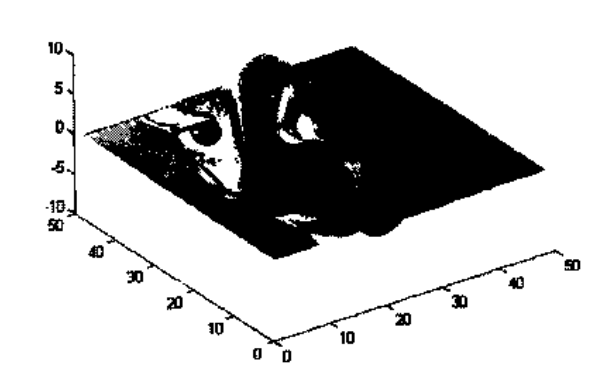


图 16-117　着色的面图

对象继承表：

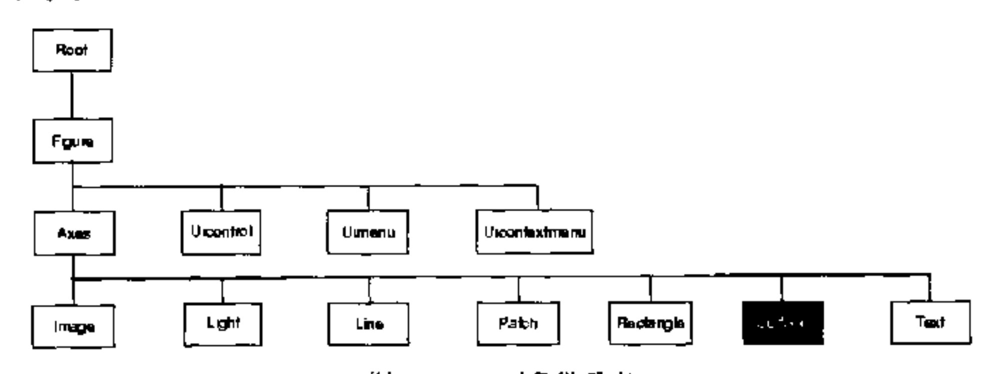


图 16-118　对象继承表

用户可以在轴上、图形窗口里和根层上设置默认的面属性：

set(0,'DefaultSurfaceProperty',PropertyValue...)

set(gcf,'DefaultSurfaceProperty',PropertyValue...)

set(gca,'DefaultSurfaceProperty',PropertyValue...)

这里，Property 面属性的名字，其默认值是用户想设置的，PropertyValue 是用户指定的值。使用 set 和 get 来可以访问面属性。

表 16-19　　面对象属性列表

属性名称	**属性描述**	**属性值**
定义对象的数据		
XData	表面的顶点的 x 坐标	值：向量或矩阵
YData	表面的顶点的 y 坐标	值：向量或矩阵
ZData	表面的顶点的 z 坐标	值：向量或矩阵
指定颜色		
CData	颜色数据	值：标量、向量或矩阵； 默认值：[](空矩阵
CDataMapping	控制 Cdata 到色图的映射	值：scaled, direct; 默认值：scaled, direct
EdgeColor	面上边的颜色	值：ColorSpec, none, flat, interp; 默认值：ColorSpec

FaceColor	面的颜色	值：ColorSpec, none, flat, interp；默认值：ColorSpec
MarkerEdgeColor	标记的颜色或用于填充标记的边的颜色	值：ColorSpec, none, auto；默认值：none
MarkerFaceColor	为有闭合形状的标记填色	值：ColorSpec, none, auto；默认值：none
控制照明的效果		
AmbientStrength	背景光的强度	值：标量≥0 和≤1；默认值：0.3
BackFaceLighting	控制指向背离照相机的面的照明	值：unlit, lit, reverselit；默认值：reverselit
DiffuseStrength	漫射光的强度	值：标量≥0 和≤1；默认值：0.6
EdgeLighting	用于照明边的方法	值：none, flat, gouraud, phong；默认值：none
FaceLighting	用于照明边的方法	值：none, flat, gouraud, phong；默认值：none
NormalMode	MATLAB 生成的或用户指定的法向向量	值：auto, manual；默认值：auto
SpecularColorReflectance	镜像反射光的混合色	值：标量 0 到 1；默认值：1
SpecularExponent	镜像反射的粗糙度	值：标量≥1；默认值：10
SpecularStrength	镜像光的强度	值：标量≥0 和≤1；默认值：0.6
VertexNormals	顶点法向向量	值：矩阵
定义边和标记		
LineStyle	从五种线样式中选择	值：-, --, :, -., none；默认值：-
LineWidth	用点数计的边的宽度	值：标量；默认值：0.5 点
Marker	在数据点绘出标记	值：见 Marker 属性；默认值：none
MarkerSize	用点数计算标记的尺寸	值：点数的大小；默认值：6
控制外观		
Clipping	剪贴	值：on, off；默认值：on
BraseMode	绘制或擦除线的方式	值：normal, none, xor, background；默认值：normal
MeshStyle	确定是否画出所有的边线	值：both, row, column；默认值：both
SelectionHighlight	高亮度显示被选中的线	值：on, off；默认值：on
Visible	线可见或不可见	值：on, off；默认值：on
控制对象的访问		
HandleVisibility	决定表面句柄是否对其它函数可视及什么时候可视	值：on, callback, off；默认值：on
HitTest	决定表面是否变为当前对象	值：on,off；默认值：on
有关调用函数执行的属性		
BusyAction	调用函数时遇到中断如何处理	值：cancel, queue；默认值：queue
ButtonDownFcn	按钮按下时执行的调用函数	值：字符串；默认值：空字符串
CreateFcn	创建表面时执行的调用函数	值：字符串；默认值：空字符串
DeleteFcn	删除表面时执行的调用函数	值：字符串；默认值：空字符串
Interruptible	执行调用函数时是否接收中断	值：on, off；默认值：on
UIContextMenu	与表面所相关的上下文菜单	值：上下文菜单的句柄
面的通用信息		
Children	没有子块的表面对象	值：[] (空矩阵)
Parent	父块对象永远是轴对象	值：轴句柄
Selected	指明表面是否处于"selected"状态	值：on、off；默认：on
Tag	用户指定的标注	值：任何字符串；默认：''(空字符串)

续 表

属性名称	属性描述	值：
Type	图形对象的类型(只读)	值：字符串'light'
UserData	用户指定的数据	值：任何矩阵； 默认：[] (空矩阵)

9. 标注文字

名称：text

标注文字。

语法：该函数有如下四种表达形式：

- text(x,y,'string')
- text(x,y,z,'string')
- text(...'PropertyName',PropertyValue...)
- h = text(...)

描述：text 是低层函数，生成文字图形对象。使用 text 函数在指定位置标注字符串。

text(x,y,'string')在点(x, y)处标注引号内的字符串。

text(x,y,z,'string')在三维坐标系内标注文字。

text(x,y,z,'string','PropertyName',PropertyValue....)在指定坐标系内的给定位置处标注设定的文字。

text('PropertyName',PropertyValue....)忽略坐标系，按照给定的属性的名称和数值标注。

h = text(..) 返回一个指向文字对象的列向量句柄，一个分量对应一个对象。

举例：

下列语句：

```
plot(0:pi/20:2*pi,sin(0:pi/20:2*pi))
text(pi,0,' \leftarrow sin(\pi)','FontSize',18)
```

在点(pi,0)处标注 sin().

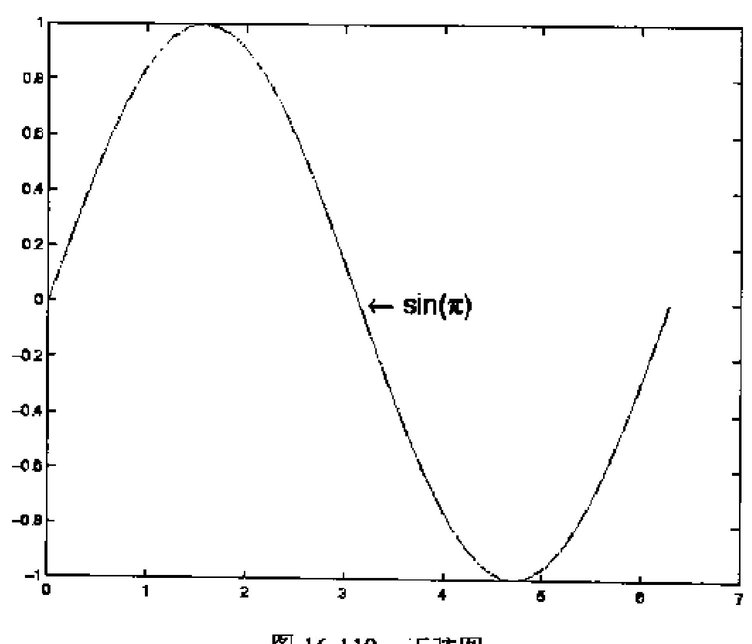


图 16-119 正弦图

对象继承表：

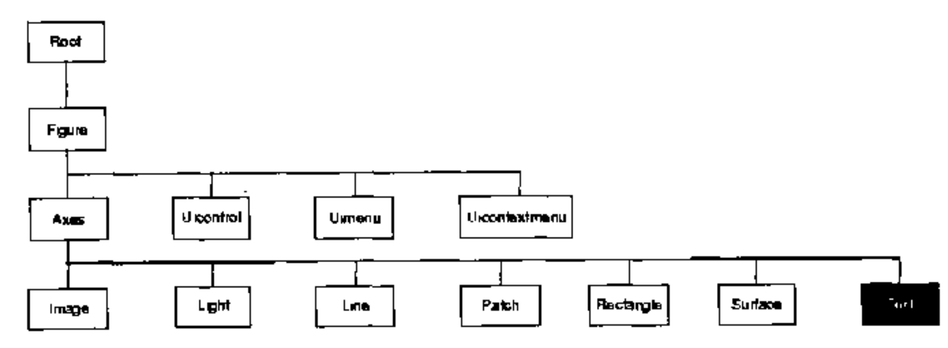


图 16-120　对象继承表

用户能在轴，图形和根的层次上设置图形属性的默认值：

set(0,'DefaulttextProperty',PropertyValue...)

set(gcf,'DefaulttextProperty',PropertyValue...)

set(gca,'DefaulttextProperty',PropertyValue...)

其中 PropertyName 是文本的属性名，PropertyValue 便是所设置的数值。使用 set 和 get 函数可设置和获取文本的值：。

属性列表。

表 16-20　　　　文本对象属性列表

属性名称	**属性描述**	**值：**
定义字符串		
Editing	控制文本是否能编辑	值：on, off；默认值：off
Interpreter	是否能用 TeX 解释	值：tex, none；默认值：tex
String	字符串	值：character string
字符串定位		
Extent	文本对象的位置和大小	值：[left, bottom, width, height]
HorizontalAlignment	文本的水平对齐方式	值：left, center, right； 默认值：left
Position	文本框的位置	值：[x, y, z] coordinates； 默认值：[] empty matrix
Rotation	文本对象的方位	值：scalar (degrees)； 默认值：0
Units	描述文本框位置属性的数值单位	值：pixels, normalized, inches, centimeters, points, data 默认值：data
VerticalAlignment	文本的垂直对齐方式	值：top, cap, middle, baseline, bottom；默认值：middle
设定字体		
FontAngle	字体是否倾斜	值：normal, italic, oblique； 默认值：normal
FontName	字体名称	值：用户系统中所安装的字体； 默认值：Helvetica 字体
FontSize	字体大小	值：一个整数默认值：10
FontUnits	描述字体大小所用的单位	值：points, normalized, inches, centimeters, pixels； 默认值：points

续 表

属性名称	属性描述	值:
FontWeight	字体是否加粗	属性值：normal, bold, light, demi；默认值：normal
控制外观		
Clipping	剪贴	值：on, off； 默认值：on
SelectionHighlight	高亮度显示被选中的文本	值：on, off； 默认值：on
Visible	文本可见或不可见	值：on, off； 默认值：on
控制对象		
HandleVisibility	控制是否能获取矩形的句柄	值：on, callback, off； 默认值：on
HitTest	控制被单击文本是否成为当前对象	值：on, off； 默认值：on
文本的通用信息		
Children	无子对象	
Parent	父对象总是轴对象	值：轴句柄标量
Selected	是否显示矩形被选中	值：on, off ； 默认值：on
Tag	用户指定的标注	值：任意字符串； 默认值：''(空串)
Type	图形对象的类型（只读）	值：字符串：'axes：'
UserData	用户指定的数据	值：任意矩阵 默认值：[] (空矩阵)
控制调用程序的执行		
BusyAction	调用函数时遇到中断如何处理	值：cancel, queue； 默认值：queue
ButtonDownFcn	按钮按下时执行的调用函数	值：字符串； 默认值：空字符串
CreateFcn	创建文本时执行的调用函数	值：字符串； 默认值：空字符串
DeleteFcn	删除文本时执行的调用函数	值：字符串； 默认值：空字符串
Interruptible	执行调用函数时是否接收中断	值：on, off； 默认值：on
UIContextMenu	文本所相关的上下文菜单	值：上下文菜单的句柄

10. 创建一个上下文菜单

名称：uicontextmenu

创建一个上下文按钮。

语法：handle = uicontextmenu('PropertyName',PropertyValue,...);

描述：uicontextmenu 创建一个上下文菜单，当用户在图形对象上右击时，它是一个能显示的菜单。

用户使用 uimenu 函数创建上下文菜单项。菜单项沿 uimenu 语句出现的顺序出现。用户使用 UIcontextMenu 属性和指定的上下文菜单句柄为属性值，将一个对象与一个文本菜单联系起来。

下表列出了对 uicontextmenu 对象十分有用的属性，并按功能将它们编组。

表 16-21　　uicontextmenu 对象属性列表

属性名称	属性描述	属性值
控制样式和外观		
Visible	用户界面上下文菜单可视化	值：on, off； 默认值：off
Position	启动可视化时的用户界面上下文菜单的位置	值：两元素向量； 默认值：[0 0]
对象的通用信息		
Children	为用户界面上下文菜单定义的用户界面菜单	值：矩阵
Parent	用户界面上下文菜单父对象	值：标量图形窗口句柄
Tag	用户指定的对象标识符	值：字符串
Type	图形对象类	值：字符串(只读)； 默认值：uicontrol
UserData	用户指定的数据	值：矩阵
控制调用函数的执行		
BusyAction	调用程序中断	值：cancel, queue； 默认值：queue
Callback	控制操作	值：字符串
CreateFcn	在创建对象过程中执行的调用程序	值：字符串
DeleteFcn	在删除对象过程中执行的调用程序	值：字符串
Interruptible	调用程序中断模式	值：on, off； 默认值：on
控制对象的访问		
HandleVisibility	句柄是否可以从命令行和 GUIs 存取	值：on, callback, off； 默认值：on

举例：

这些语句定义了一个与一条线有关的上下文菜单。当用户在线上的任意位置单击右键时，就会出现菜单。菜单项可以使用户改变线的样式。

```
% Define the context menu
cmenu = uicontextmenu;
% Define the line and associate it with the context menu
hline = plot(1:10, 'UIContextMenu', cmenu);
% Define callbacks for context menu items
cb1 = ['set(hline, "LineStyle", "--")'];
cb2 = ['set(hline, "LineStyle", ":")'];
cb3 = ['set(hline, "LineStyle", "-")'];
% Define the context menu items
item1 = uimenu(cmenu, 'Label', 'dashed', 'Callback', cb1);
item2 = uimenu(cmenu, 'Label', 'dotted', 'Callback', cb2);
item3 = uimenu(cmenu, 'Label', 'solid', 'Callback', cb3)
```

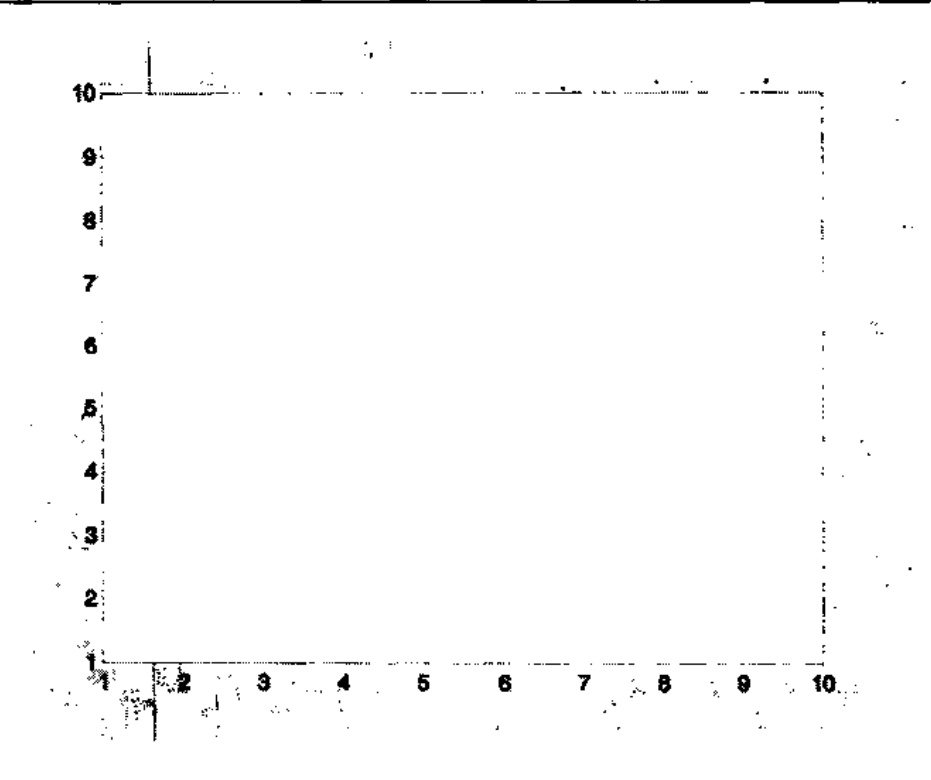


图 16-121　与线有关的上下文菜单

当用户在线上单击右键时，就出现上下文菜单。

对象继承表：

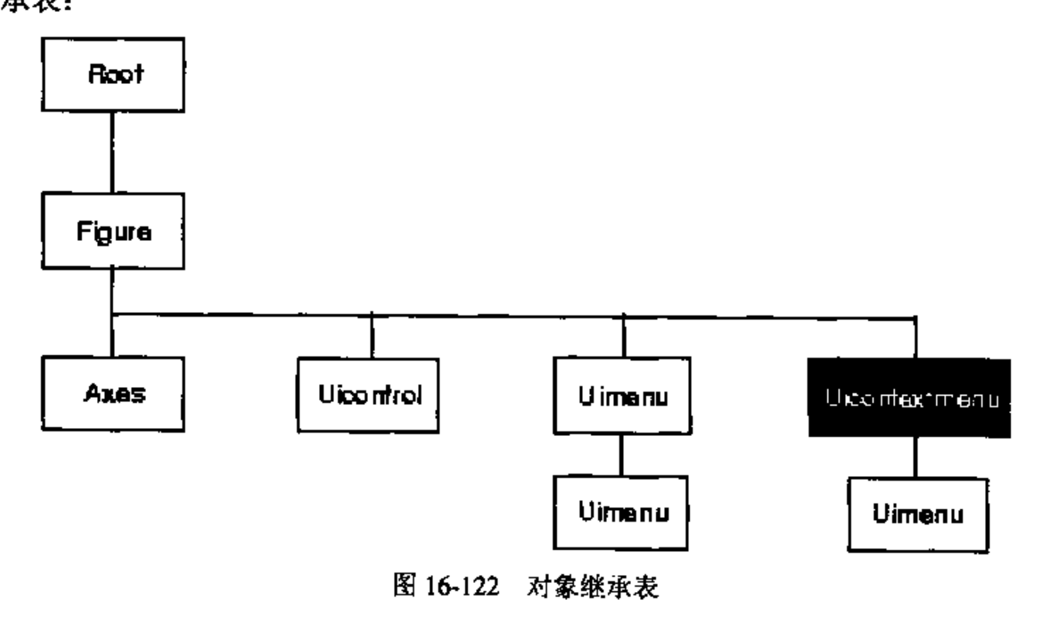


图 16-122　对象继承表

16.14　图像处理（图像窗口）

1. 捕捉

名称：capture

capture 函数在 Release 11 (5.3)版本中已经是要淘汰的函数，getframe 提供了同样的功能，

并且支持真彩色。

语法：该函数有如下三种表达形式：

- capture
- capture(h)
- [X,cmap] = capture(h)

描述：capture 生成一个当前图形的位图拷贝，包括所有的 uicontrol 图形对象。它在一个新的图形窗口把该拷贝显示出来。

capture(h) 生成一个由 h 指定的图形对象所复制的新图形。

[X,cmap] = capture(h)返回一个图形的数据矩阵 X 和一个色图。用户可以通过下面的语句显示所得到的信息：

```
colormap(cmap)
image(X)
```

2. 清除当前的图形窗口

名称：clf

清除当前的图形窗口。

语法：该函数有如下两种表达形式：

- clf
- clf reset

描述：clf 删除当前图形中所有未隐藏的对象（即它们的 HandleVisibility 值为 on）。

clf reset 删除当前图形中所有的对象，不管它们的 HandleVisibility 属性的取值，并且除 Position、Units、PaperPosition 和 PaperUnits 取默认值以外，重置其它所有的图形属性。

3. 清除窗口里的命令

名称：clc

清除窗口里的命令。

语法：clc

描述：clc 清除窗口里的命令。

举例：

在命令窗口的相同位置处显示一列随机矩阵。

```
clc
for i =1:25
     home
     A = rand(5)
end
```

4. 删除指定的图形

名称：close

删除指定的图形。

语法：该函数有如下六种表达形式：

- close
- close(h)

- close name
- close all
- close all hidden
- status = close(...)

描述：close 删除当前的图形或指定的图形。它随意返回 close 操作的状态。

close 删除当前的图形(等效于 close(gcf))。

close(h)删除由 h 确定的图形。如果 h 是一个向量或矩阵，close 删除由 h 确定的所有图形。

close name 删除带有指定名字的图形。

close all 删除所有句柄没有消隐的图形。

close all hidden 删除所有带有消隐句柄的图形

status = close(...)如果指定的窗口被删除，返回 1，反之，则是 0。

5. 获取当前图形的句柄

名称：gcf

获取当前图形的句柄。

语法：h = gcf

描述：h = gcf 返回当前图形窗口的句柄，所谓当前图形是指图形窗口中由绘图指令如 plot, title,和 surf 等函数的结果。如果该句柄不存在，MATLAB 会自动生成一个并返回它的句柄。如果用户不希望 MATLAB 自动生成当前不存在的句柄，可以使用下列语句：

get(0,'CurrentFigure')

6. 重新绘制当前图形

名称：refresh

重新绘制当前图形。

语法：该函数有如下种表达形式：

refresh

refresh(h)

描述：refresh 擦除当前图形并重新绘制。

refresh(h) 重新绘制句柄 h 所指向的图形。

7. 确定所画图形对象的位置

名称：newplot

确定所画图形对象的位置。

语法：该函数有如下种表达形式

- newplot
- h = newplot

描述：newplot 为后续的图形命令准备一个图形窗口和轴。

h = newplot 为后续的图形命令准备一个图形窗口和轴，并返回指向当前轴的一个句柄。图形窗口和轴的 NextPlot 属性决定 Nextplot 如何工作。下面两个表描述了不同属性值的工作情况。

首先，newplot 读出当前图形窗口的 NextPlot 属性并激活它们；

表 16-22　　　　　　　　　　图形窗口对 NextPlot 的影响

NextPlot	发生
add	向没有清除已存在图形对象的当前图形对象窗口绘制图形
replacechildren	删除所有其 HandleVisibility 属性被设置为 on 的子对象，并重新设置要添加的图形窗口 NextPlot 属性。 它清除当前的图形窗口，与发布 clf 命令等效。
replace	删除所有的子对象(忽视 HandleVisibility 属性的设置)并重新将图形窗口属性设置为默认，除了下面的情况：在忽视由用户定义默认值的情况下，NextPlot 被重新设置。Position、Units、PaperPosition 和 PaperUnits 不被重新设置。 它清除和重新设置当前的图形窗口，与发布 clf reset 命令等效

在 newplot 建立了将在哪一个图形窗口里绘图后，它读出当前轴的 NextPlot 属性并激活它们。

表 16-23　　　　　　　　　　轴对 NextPlot 的影响

NextPlot	描述
add	在当前轴里绘制图形，保留所有已经存在的图形
replacechildren	删除所有其 HandleVisibility 属性被设置为 on 的子对象，但是不重新设置轴属性。 它像 clf 命令一样清除当前轴。
replace	删除所有的子对象(忽视 HandleVisibility 属性的设置)并重新将轴属性设置为默认，除了 Position 和 Units 以外。 它像 clf reset 命令一样清除和重新设置当前轴.

16.15　图像处理（坐标轴）

1. 轴的缩放比例和外观

名称：axis

轴的缩放比例和外观。

语法：该函数有如下十七种表达形式：

- axis([xmin xmax ymin ymax])
- axis([xmin xmax ymin ymax zmin zmax])
- v = axis
- axis auto
- axis manual
- axis tight
- axis fill
- axis ij
- axis xy
- axis equal
- axis image
- axis square
- axis vis3d

- axis normal
- axis off
- axis on
- [mode,visibility,direction] = axis('state')

描述：axis([xmin xmax ymin ymax])设置当前二维图形坐标轴的刻度范围。

axis([xmin xmax ymin ymax zmin zmax])设置当前三维图形坐标轴的刻度范围。

v = axis 返回一个存有 x、y、z 轴的缩放比例的行向量。当前坐标轴是二维或三维决定 v 是四维或六维向量。返回的数值是当前轴的 XLim、Ylim 和 ZLim 属性的取值。

axis auto 设置当前坐标轴为默认状态。用户可以限制自动功能只对指定的坐标轴有效。例如，axis 'auto x'只对 x 轴设置为默认状态。

axis manual and axis(axis)冻结和保持当前轴的缩放比例。所以如果 hold 开关是打开的，则下面的图形也使用相同的刻度范围。该函数设置属性 XLimMode、YLimMode 和 ZLimMode 的取值为 manual。

axis tight 设置纵横比使得数据在各个方向上的单位相同。这不同于 axis equal，因为绘图框的纵横比是自动调整的。

axis fill 设置轴的刻度范围在数据范围内。

axis ij 设置坐标系的原点在左上角。i 轴垂直，从上到下数值增加；j 轴水平，从左到右数值增加。

axis xy 设置为通常的左下角为原点的坐标系。x 轴水平，数值从左到右增加；y 轴垂直，数值从下到上增加。

axis equal 数值纵横比，使得各个方向上的数据单位相同。x，y，z 轴的缩放比例根据 x，y，z 各个方向上的数据范围自动调整。

axis image 除了绘图框固定外，和 axis equal 函数的效果相同。

axis square 使得当前各个轴的长度相同，至于各个轴的缩放比例则各自调整。

axis vis3d 冻结纵横比；

axis normal 自动调整轴的纵横比和数据单位的纵横比。

axis off 关闭坐标轴的显现状态。

axis on 打开坐标轴的显现状态。

[mode,visibility,direction] = axis('state')返回当前坐标轴的属性设置到三个字符串中：

表 16-24　　　　返回当前坐标轴的属性表

输出参数	返回的字符串
Mode	'auto' | 'manual'
Visibility	'on' | 'off'
Direction	'xy' | 'ij'

如果 XLimMode、YLimMode 和 ZLimMode 是 auto，则 mode 是 auto；如果 XLimMode、YLimMode 或 ZLimMode 是 manual，则 mode 是 manual。

举例：

(1) 下列语句：

```
x = 0:.025:pi/2;
plot(x,tan(x),'-ro')
```

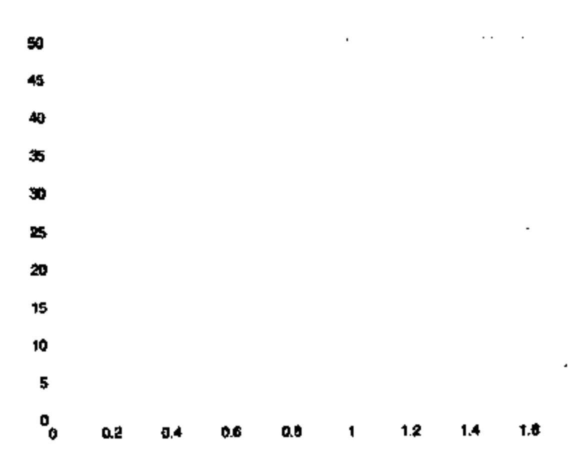


图 16-123 设置纵横比的图形

使用自动缩放比例，因为 y 的最大值是 tan(1.57),大大超过了 1000：

(2) 再输入下面的语句，使得右边的图形显示了更好的效果：

axis([0 pi/2 0 5])

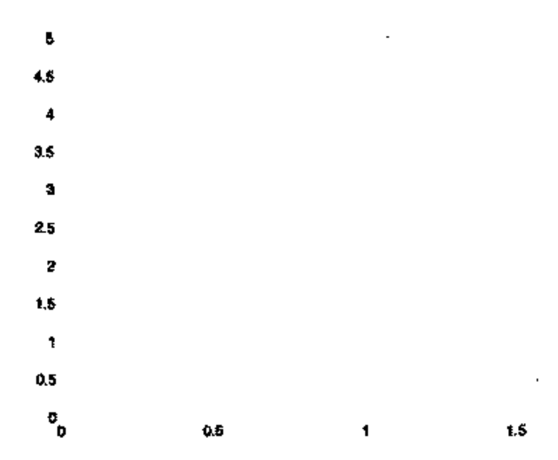


图 16-124 设置纵横比的图形

下表显示了 axis equal，axis normal，axis square，和 axis image 所设置的坐标轴的属性的取值：

表 16-25 坐标轴属性取值表

轴的属性	axis equal	axis normal	Axis square	axis tightequal
DataAspectRatio	[1 1 1]	没有设置	没有设置	[1 1 1]
DataAspectRatioMode	manua	auto	Auto	manua
PlotBoxAspectRatio	[3 4 4]	not set	[1 1 1]	auto
PlotBoxAspectRatioMode	manual	auto	Manual	auto
Stretch-to-fill	禁止	有效	禁止	禁止

2. 清除当前轴

名称：cla

清除当前轴。

语法：该函数有如下两种表达形式：

- cla
- cla reset

描述：cla 从当前轴删除所有的没有消隐的句柄的图形对象。

cla reset 从当前轴删除所有的图形对象，忽略它们的 HandleVisibility 属性的设置，除了 Position 和 Units 外，重新将所有的轴属性设置为它们的默认值，

3. 获取当前轴的句柄

名称：gca

获取当前轴的句柄。

语法：h = gca

描述：h = gca 返回指向当前图形中当前轴的句柄。如果轴不存在，MATLAB 生成一个新的轴并返回该轴的句柄。如果用户不希望 MATLAB 在轴不存在时自动生成一个新轴，可以使用下面的语句：

get(gcf,'CurrentAxes')

16.16 对象操作

1. 将图形对象重新设置为它们的默认值

名称：reset

将图形对象重新设置为它们的默认值。

语法：reset(h)

描述：reset(h)在由 h 确定的对象上将所有具有 factory 的属性重新设置为默认。使用下面的语句可以看到 factory 默认列表，

get(0,'factory')

如果 h 是一个图形，MATLAB 将不重新设置 Position、Units、PaperPosition 和 PaperUnits。

如果 h 是一个轴，MATLAB 将不重新设置 Position 和 Units。

举例：

reset(gca) 重新设置当前轴的属性。

reset(gcf) 重新设置当前图形的属性。

2. 选择、移动、调整和复制轴和用户界面控制图形对象

名称：selectmoveresize

选择、移动、调整和复制轴和用户界面控制图形对象。

语法：该函数有如下两种表达形式：

- A = selectmoveresize;
- set(h,'ButtonDownFcn','selectmoveresize')

描述：selectmoveresize 可以用于作为轴按钮和用户界面控制按钮沿 functions 的返回路线。当它被执行时，它选择对象并允许用户移动、调整和复制它。

3. 使用鼠标旋转轴

名称：rotate3d

使用鼠标旋转轴。

语法：该函数有如下三种表达形式：

- rotate3d
- rotate3d on
- rotate3d off

描述：rotate3d on 使得用户可以使用鼠标旋转当前图形中的坐标轴。当用鼠标旋转图形的功能允许时，单击图形并拖动鼠标，图形跟随鼠标的拖动而动态地旋转，直到释放鼠标。

rotate3d off 使鼠标旋转当前图形的功能关闭。

rotate3d toggles 切换鼠标旋转图形功能开闭状态。

双击图形恢复原始的视图。

16.17　用户交互输入

1. 使用鼠标输入数据

名称：ginput

使用鼠标输入数据。

语法：该函数有如下三种表达形式：

- [x,y] = ginput(n)
- [x,y] = ginput
- [x,y,button] = ginput(...)

描述：ginput 允许用户使用鼠标或光标键选取图形上的点。在 ginput 函数能接收输入时，该图形窗口必须是激活状态。

[x,y] = ginput(n)使得用户可以把所选取的 n 个点的 xy 坐标相应放在列向量 x 和 y 中。在输入 n 个数据点前，按下回车键可终止输入。

[x,y] = ginput 可接收无限的数据点，直到用户按下回车键才终止接收。

[x,y,button] = ginput(...)返回点的 x，y 坐标和用户所按下的鼠标键或键盘的键值，对于鼠标，1 表示左键，2 表示中间键，3 表示右键；对于键盘，则返回用户所按下的键的 ASCII 值。

举例：

从当前二维图形窗口选取 10 个点：

[x,y] = ginput(10)

2. 在二维图形上进行放大和缩小

名称：zoom

在二维图形上进行放大和缩小.

语法：该函数有如下九种表达形式：

- zoom on

- zoom off
- zoom out
- zoom reset
- zoom
- zoom xon
- zoom yon
- zoom(factor)
- zoom(fig, option)

描述：zoon on 开启交互式缩放的功能。在一个图形窗口里使用交互式缩放，当用户的指针位于一个轴内时，按下鼠标按钮将缩放鼠标下面的点。缩放改变了轴的范围。对于单按钮鼠标，通过按下鼠标按钮来进行放大，通过同时按 Shift 键和鼠标按钮来进行缩小。对于双按钮或三按钮鼠标，通过按下鼠标左按钮来进行放大，通过按下鼠标右按钮来进行缩小

当交互式缩放功能开启时，在一个轴上单击和拖动就可以画出一个橡皮框(rubber-band)，当放开鼠标按钮时，轴就放大到由橡皮框围成的区域。

在轴上双击将轴返回至其初始缩放设置。

zoon off 关闭交互式缩放。

zoom out 将图形返回至其初始缩放设置。

zoom reset 将当前的缩放设置记为初始的缩放设置。当交互式缩放模式启动时，后续的对缩小进行调用或双击，将返回这个缩放水平。

zoom 切换交互式缩放状态。

zoom xon 和 zoom yon 分别为 x 轴和 y 轴设置缩放启动。

zoom(factor)通过指定的缩放系数来进行缩放，它并不影响交互式缩放模式。系数大于 1 就放大 factor 倍，当数字是大于 0 而小于 1 时，缩小 1/factor。

zoom(fig, option)除了在使用这个语法的当前图形上外，上面的任何选项都可以在一个图形上被指定。

16.18 关注区域处理

1. *用鼠标拖动矩形*

名称：dragrect

用鼠标拖动矩形。

语法：该函数有如下两种表达形式：

- [finalrect] = dragrect(initialrect)
- [finalrect] = dragrect(initialrect,stepsize)

描述：[finalrect] = dragrect(initialrect)跟踪一个或多个屏幕上的矩形。定义 n×4 的矩阵 rect 存放矩形。矩阵的每一行包含矩形的初始位置：数据格式[left bottom width height]。dragrect 函数返回矩形的最后位置到矩阵 finalrect 中。

[finalrect] = dragrect(initialrect,stepsize)按照 stepsize 指定的增量移动矩形。第一个矩形的

左下角被局限在按 stepsize 的大小所定义的坐标网格线上移动，并且以后的矩形也按照初始的增量移动。当鼠标释放后，[finalrect] = dragrect(...)返回最后矩形的位置。默认的 stepsize 值是 1。

举例：

拖动一个宽 50，高 100 个像素的矩形：

```
waitforbuttonpress
point1 = get(gcf,'CurrentPoint') % button down detected
rect = [point1(1,1) point1(1,2) 50 100]
[r2] = dragrect(rect,100)
```

2. 完成待解决的绘图事件

名称：drawnow

完成等待的绘图事件。

语法：drawnow

描述：drawnow 刷新事件队列并更新图形窗口。

举例：

执行下面的语句：

```
x = -pi:pi/20:pi;
plot(x,cos(x))
drawnow
title('A Short Title')
grid on
```

当执行 drawnow 功能并执行最后的语句后，作为一个 M 文件，更新当前的图形。

第十七章　创建用户图形界面

17.1　对话框

1. 创建对话框

名称：dialog

创建对话框。

语法：h = dialog('PropertyName',PropertyValue,...)

描述：h = dialog('PropertyName',PropertyValue,...)返回一个对话框句柄。这个函数产生一个图形对象并且为对话框设置推荐的图形属性。你可以指定任何正确的图形属性值。

2. 创建错误对话框

名称：errordlg

创建错误对话框。

语法：该函数有如下种表达形式：

- errordlg
- errordlg('errorstring')
- errordlg('errorstring','dlgname')
- errordlg('errorstring','dlgname','on')
- h = errordlg(...)

描述：errordlg 生成一个出错对话框，或者如果指定的对话框存在时，errordlg 将指定的对话框调到其它窗口前台。

errordlg 显示一个名称为'Error Dialog'的对话框，这个对话框中包含字符串'This is the default error string.'。

errordlg('errorstring')显示一个包含字符串'errorstring'的名为'Error Dialog'的对话框。

errordlg('errorstring','dlgname') 显示一个包含字符串'errorstring'的名为'dlgname'的对话框。

errordlg('errorstring','dlgname','on')指定是否取代一个已经存在的有相同名字的对话框。'on'把一个有相同名字的出错对话框提到前台。在这种情况下，errordlg 没有生成新的对话框。

h = errordlg(...)返回一个对话框的句柄。

MATLAB 改变对话框的大小来适应字符串'errorstring'的长度。这个出错对话框有一个 OK 按钮，在你按下 OK 按钮或者 Return 键之前，此对话框将一直保留在屏幕上。在按了按钮以后，出错对话框消失。

对话框的外观依赖于你所使用的视窗系统。

举例：

函数

errordlg('File not found','File Error');

在一个 UNIX 系统中显示如下的对话框：

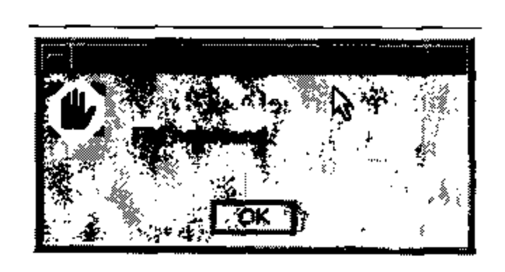


图 17-1 错误对话框示例

3. 显示帮助对话框

名称：helpdlg

显示帮助对话框。

语法：该函数有如下种表达形式：

- helpdlg
- helpdlg('helpstring')
- helpdlg('helpstring','dlgname')
- h = helpdlg(...)

描述：helpdlg 产生一个帮助对话框或者把一个指定的帮助对话框提到前台。

helpdlg 显示包含字符串 This is the default help string.'的名为'Help Dialog'的对话框。

helpdlg('helpstring')显示包含由'helpstring'指定的字符串的名为'Help Dialog'的对话框。

helpdlg('helpstring','dlgname') 显示包含字符串'helpstring'的名为'dlgname'的对话框。

h = helpdlg(...)返回一个对话框句柄。

MATLAB 将'helpstring'中的内容自动换行来适应一个对话框的宽度。对话框将一直保留在屏幕上直到你按 OK 按钮或 Return 键。在按了按钮后，帮助对话框消失。

举例：

表达式

helpdlg('Choose 10 points from the figure','Point Selection');

显示如下的对话框：

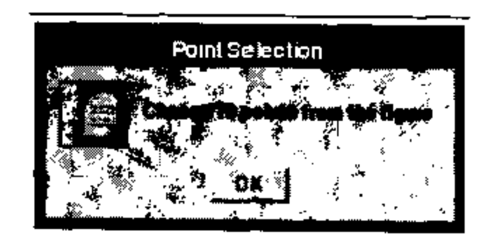


图 17-2 帮助对话框示例

4. 创建输入对话框

名称：inputdlg

创建输入对话框。

语法：该函数有如下种表达形式：

- answer = inputdlg(prompt)
- answer = inputdlg(prompt,title)
- answer = inputdlg(prompt,title,lineNo)
- answer = inputdlg(prompt,title,lineNo,defAns)
- answer = inputdlg(prompt,title,lineNo,defAns,Resize)

描述：answer = inputdlg(prompt)创建一个模式对话框并返回用户在数组中输入的内容。prompt 是一个包含提示字符串的数组。

answer = inputdlg(prompt,title)中，title 为对话框指定一个标题。

answer = inputdlg(prompt,title,lineNo)中，lineNo 为用户的每个输入值指定输入行数。lineNo 可以是标量、列向量或者矩阵。如果 lineNo 是一个标量，它适用于所有的提示字符串。如果 lineNo 是一个列向量，则向量的每个元素为一个提示符指定输入时的行数。如果 lineNo 是一个矩阵，它的尺寸应该是 m×2。这里 m 是对话框中提示符的个数。每一行对应一个提示符。第一列指定输入的行数。第二列指定字符的域宽。

answer = inputdlg(prompt,title,lineNo,defAns)中，defAns 指定每个提示符显示的默认值。defAns 必须和 prompt 有同样数目的元素并且所有的元素必须是字符串。

answer = inputdlg(prompt,title,lineNo,defAns,Resize)中，Resize 说明对话框是否改变尺寸。允许值是'on'和'off'，'on'意味着对话框可以改变尺寸而并非固定模式。

举例：

下面的表达式创建了一个输入整数和色图名称的对话框，每个值允许输入一行。

```
prompt     = {'Enter matrix size:','Enter colormap name:'};
title      = 'Input for peaks function';
lines = 1;
def        = {'20','hsv'};
answer     = inputdlg(prompt,title,lines,def);
```

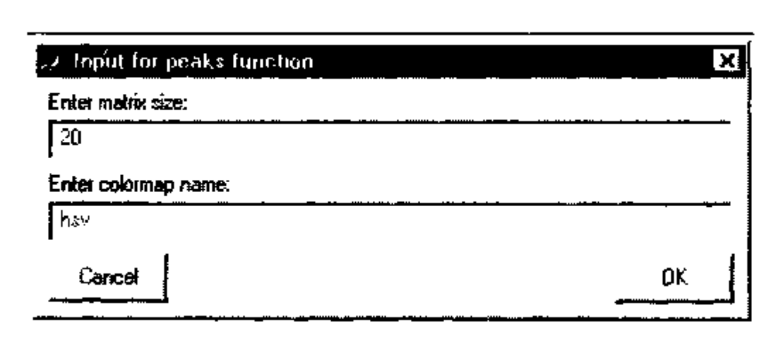


图 17-3　输入对话框示例

5. 创建选择列表内容的对话框

名称：listdlg

创建选择列表内容的对话框

语法：[Selection,ok] = listdlg('ListString',S,...)

描述：[Selection,ok] = listdlg('ListString',S)生成一个可以使用户从一个列表中选择一个

或者多个选项的模式对话框。Selection 是已选择的字符串的索引向量（在只有一个选项的模式下，它的长度为 1）。当 ok 是 0 时，Selection 是空向量[]。当你点击 OK 按钮时，ok 是 1；而当你点击 Cancel 按钮或者关闭对话框时，ok 是 0。双击一个选项或者在多项被选定后按 Return 键都相当于按 OK 按钮。对话框还有一个 Select all 的按钮（在多选择模式下），使你能够选中所有的选项。

输入以参数和值的形式列出如下：

表 17-1　　列表框中的参数列表

参数	描述
'ListString'	指定列表框选项的字符串数组
'SelectionMode'	指定是否一个或者多个选项可被选择的字符串，值为'single'或者'multiple'。默认值为'multiple'。
'ListSize'	用像素衡量的列表框的尺寸，用一个二元素向量来指定，[宽度 高度]。默认值为[160 300]
'InitialValue'	最初被选择的列表框选项的索引向量。默认值为 1，即第一项。
'Name'	对话框标题的字符串，默认值是''，即空字符串。
'PromptString'	在列表框上部的正文中出现的字符串的数组或矩阵。默认值为{}。
'OKString'	指定 OK 按钮的字符串。默认值为'OK'。
'CancelString'	指定 Cancel 按钮的字符串。默认值为'Cancel'
'uh'	用户界面控制按钮的高度，用像素表示。默认值为 18。
'fus'	框架和用户截面控制按钮的间隔，用像素表示。默认值为 8。
'ffs'	框架数字的间隔，用像素来表示。默认值为 8。

举例：

这个例子显示一个可以使用户从当前目录中选择一个文件的对话框。函数将返回一个向量。该向量的第一个元素是被选文件的索引；而第二个元素在没有做选择时为 0，在做了选择后为 1。

```
d = dir;
str = {d.name};
[s,v] = listdlg('PromptString','Select a file:',...
                'SelectionMode','single',...
                'ListString',str)
```

6. 创建消息对话框

名称： msgbox

创建消息对话框。

语法： 该函数有如下种表达形式：

- msgbox(message)
- msgbox(message,title)
- msgbox(message,title,'icon')
- msgbox(message,title,'custom',iconData,iconCmap)
- msgbox(...,'createMode')
- h = msgbox(...)

描述： msgbox(message)产生一个消息对话框，其中的消息自动进行换行来适应具有适当尺寸的图框。message 是一个字符串向量，字符串矩阵或数组。

msgbox(message,title)为消息对话框指定标题。

msgbox(message,title,'icon')指定在消息对话框中显示哪一个图标。'icon'可以是'none'、'error'、'help'、'warn'或者是'custom'。默认值是'none'。

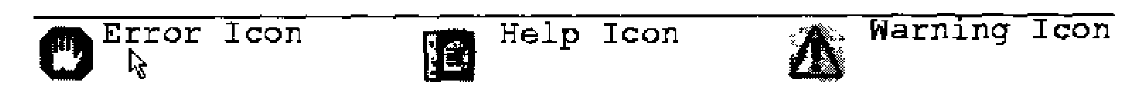


图 17-4 消息对话框的图标示例

msgbox(message,title,'custom',iconData,iconCmap)定义一个用户定制的图标。iconData 包含着定义图标的图形数据；iconCmap 是图形所用的色图。

msgbox(...,'createMode')指定消息框是否为模式化的，如果是非模式化的，是否要取代其它有同样标题的消息框。'createMode'的正确值是'modal'，'non-modal'和'replace'。

h = msgbox(...)返回一个对话框的句柄 h，它是一个图形对象的句柄。

7. 显示页面的版面对话框

名称：pagedlg

显示页面的版面对话框。

语法：该函数有如下种表达形式：

- pagedlg
- pagedlg(fig)

描述：pagedlg 对当前图形显示一个页面的版面对话框。这个对话框使用户可以进行页面设置。

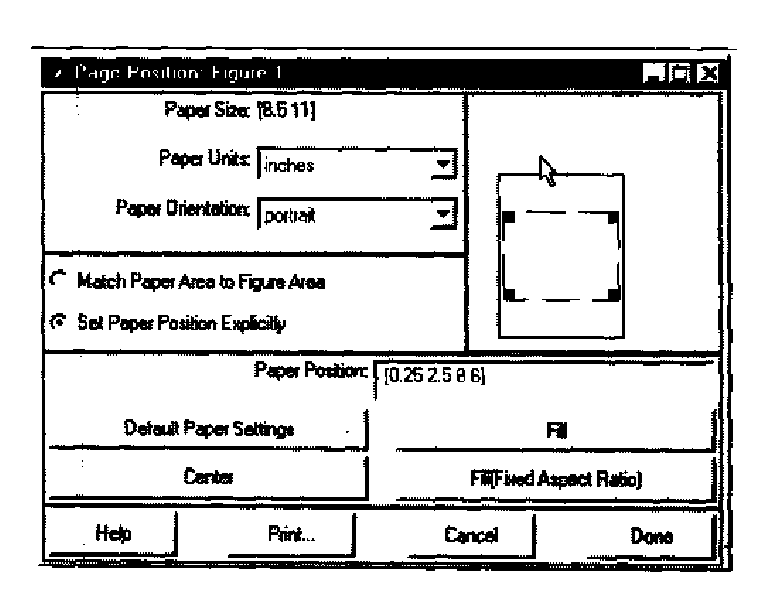


图 17-5 页面设置对话框

pagedlg(fig)对由 fig 指定的图形显示一个页面设置对话框。

8. 显示打印对话框

名称：printdlg

显示打印对话框。

语法：该函数有如下种表达形式：

- printdlg

- printdlg(fig)
- printdlg('-crossplatform',fig)

描述：printdlg 打印当前的图形。

printdlg(fig)生成一个对话框，由此用户可以打印句柄 fig 指定的图形窗口。注意此命令不能打印用户菜单。

printdlg('-crossplatform',fig)对微软的 Windows 操作系统和 Macintosh 机显示一种标准的交叉平台式的 MATLAB 打印对话框，而不是内置式的打印对话框。但要把此选项插在参量 fig 之前。

9. 创建问题对话框

名称：questdlg

创建问题对话框。

语法：该函数有如下种表达形式：

- button = questdlg('qstring')
- button = questdlg('qstring','title')
- button = questdlg('qstring','title','default')
- button = questdlg('qstring','title','str1','str2','default')
- button = questdlg('qstring','title','str1','str2','str3','default')

描述：button = questdlg('qstring')显示一个提出问题'qstring'的模式对话框。这个对话框有三个默认按钮：Yes、No 和 Cancel。'qstring'是一个数组或字符串，可以自动换行以适应对话框。button 包含点中的按钮的名字。

button = questdlg('qstring','title')显示一个在对话框标题条中显示'title'的问询对话框。

button = questdlg('qstring','title','default')指定在按 Return 键时哪个按钮是默认值。'default'一定是'Yes'，'No'或者'Cancel'。

button = questdlg('qstring','title','str1','str2','default')生成一个包含两个标示为'str1'和'str2'的按钮的问询对话框。'default'指定默认的按钮选择，并且必须是'str1'或者'str2'之一。

button = questdlg('qstring','title','str1','str2','str3','default')生成一个包含三个标示为'str1'、'str2'和'str3'的按钮的问询对话框。'default'指定默认的按钮选择，并且必须是'str1'、'str2'或者'str3'之一。

举例：

下例生成一个问询对话框询问用户是否继续一个假定的操作：

```
button = questdlg('Do you want to continue?',...
'Continue Operation','Yes','No','Help','No');
if strcmp(button,'Yes')
    disp('Creating file')
elseif strcmp(button,'No')
    disp('Canceled file operation')
elseif strcmp(button,'Help')
    disp('Sorry, no help available')
end
```

10. 为读入而显示用于检索文件名的对话框

名称：uigetfile

为读入而显示用于检索文件名的对话框。

语法：该函数有如下种表达形式：

- uigetfile
- uigetfile('FilterSpec')
- uigetfile('FilterSpec','DialogTitle')
- uigetfile('FilterSpec','DialogTitle',x,y)
- [fname,pname] = uigetfile(...)

描述：uigetfile 显示一个用于检索一个文件的对话框。这个对话框列出了当前目录中的文件和目录。

uigetfile('FilterSpec')显示了一个列出当前目录中所有文件的对话框。FilterSpec 确定最先显示的文章，它可以是文件的全名也可以包含通配符*。例如，'*.m'生成一个对话框的列表，只显示 MATLAB 中的 M 文件。

uigetfile('FilterSpec','DialogTitle')显示一个标题为 DialogTitle 的对话框。

uigetfile('FilterSpec','DialogTitle',x,y)把对话框定位在[x,y]，这里 x 和 y 是距屏幕左边界和上边界的距离，以像素为单位。注意到一些平台不支持对话框的定位。

[fname,pname] = uigetfile(...)返回在对话框中选择的文件的名称和路径。你点击 Done 按钮后，fname 保存被选中的文件的名称，而 pname 保存路径名。如果你点 Cancel 按钮或者发生错误时，fname 和 pname 都会被设置为 0。

举例：

下面的表达式显示一个可以使用户找到一个文件的对话框。表达式列出了一个被选中目录中所有的 MATLAB 的 M 文件。选中文件的名称和目录用 fname 和 pname 返回。

[fname,pname] = uigetfile('*.m','Sample Dialog Box')

对话框的确切形式有赖于你所用的操作系统。

11. 为写入而显示用于检索文件名的对话框

名称：uiputfile

为写入而显示用于检索文件名的对话框。

语法：该函数有如下种表达形式：

- uiputfile
- uiputfile('InitFile')
- uiputfile('InitFile','DialogTitle')
- uiputfile('InitFile','DialogTitle',x,y)
- [fname,pname] = uiputfile(...)

描述：uiputfile 显示一个可写入文件名称的对话框。这个对话框列出当前目录中的文件和目录。

uiputfile('InitFile')显示的对话框包含当前路径中由 InitFile 确定的文件的列表。InitFile 可以是一个文件的全名，也可以包含通配符*。例如，指定'*.m'（默认值）可以生成一个只显示 MATLAB 中 M 文件的对话框。

uiputfile('InitFile','DialogTitle')显示一个标题为 DialogTitle 的对话框。

uiputfile('InitFile','DialogTitle',x,y) 把对话框定位在屏幕上[x,y]处，这里 x 和 y 是距屏幕左边界和上边界的距离，以像素为单位。注意到一些平台不支持对话框的定位。

[fname,pname] = uiputfile(...)返回在对话框中选择的文件的名称和路径。如果你点击 Cancel 按钮或者发生错误时，fname 和 pname 都会被设置为 0。

举例：

下面的表达式显示一个标题为'Save file name'，文件名为 animinit.m 的对话框。对话框的确切形式依赖于所用的操作系统。

[newfile,newpath] = uiputfile('animinit.m','Save file name');

在 Windows 系统中为如下形式：

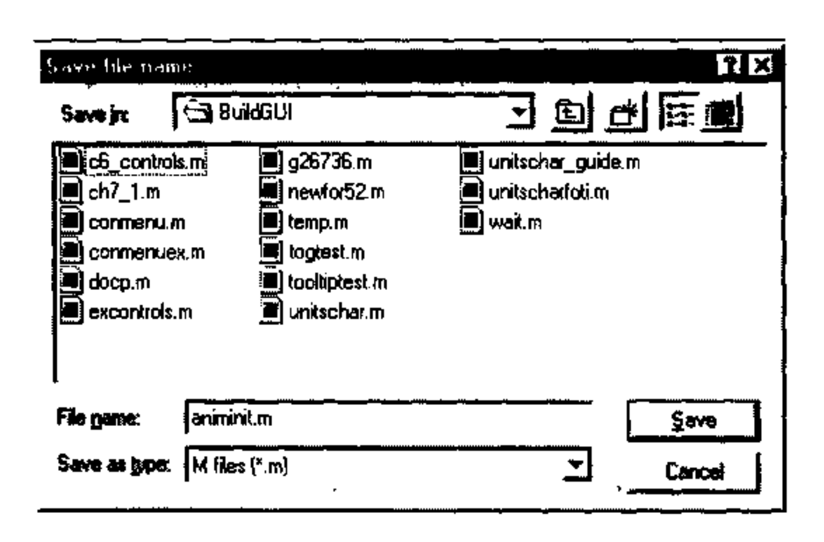


图 17-6 保存文件对话框

12. 从对话框里交互式地设置对象的 ColorSpec

名称：uisetcolor

从对话框里交互式地设置对象的 ColorSpec。

语法：c = uisetcolor(h_or_c, 'DialogTitle');

描述：uisetcolor 显示一个用于用户填写的对话框，然后把所选的颜色用于第一个量所指定的图形对象的适当的性质。

h_or_c 是一个图形对象的句柄或者是一个 RGB 三元组。如果你指定一个句柄，它必须是一个有颜色特性的图形目标。如果你指定一种颜色，那么它必须是一个正确的 RGB 三元组（例如，对红色是[1 0 0]）。指定的颜色用于初始化对话框。如果没有指定初始化的颜色，对话框初始化为黑色。

DialogTitle 是一个用于对话框标题的字符串。

c 是用户选择的 RGB 值。当用户点击对话框中 Cancel 键或者发生错误时，如果有提供值的话，c 被设置为输入的 RGB 三元组，否则被设置为 0。

13. 交互式修改对象的字体特征

名称：uisetfont

交互式修改对象的字体特征。

语法：该函数有如下种表达形式：

- uisetfont
- uisetfont(h)
- uisetfont(S)
- uisetfont(h,'DialogTitle')
- uisetfont(S,'DialogTitle')
- S = uisetfont(...)

描述：uisetfont 对于一个 text，axes 或者 uicontrol 对象可以改变字体的性质(FontName，FontUnits，FontSize，FontWeight 和 FontAngle)。函数返回一个由字体性质和值所组成的结构体。用户可以预先给对话框指定一个名称。

uisetfont 显示一个对话框并且返回被选择的字体性质。

uisetfont(h)显示一个对话框，用句柄为 h 的对象的性质初始化字体的性质。所选的字体的性质值被用于当前的对象。如果提供第二个参量的话，它为对话框指定一个名称。

uisetfont(S)显示一个对话框，用指定的结构体 S 定义的值来初始化字体性质。S 必须为 FontName，FontUnits，FontSize，FontWeight 和 FontAngle 中的一个或多个特性定义合法的值，并且域名和性质名必须相匹配。在定义其它性质时，它们可以被忽略。如果提供第二个参量，它为对话框指定一个名称。

uisetfont('DialogTitle')显示一个标题为 DialogTitle 的对话框并且返回对话框中所选的字体性质的值。

如果指定了左侧的量，则 FontName，FontUnits，FontSize，FontWeight 和 FontAngle 返回结构体中各域的值。如果用户选择对话框中的 Cancel 或发生错误时，输出的值全部设置为 0。

举例：

下面的表达式能生成一个 text 对象，然后显示一个标示为 Revise Font 的对话框使你能够改变字体的特征。

```
h = text(.5,.5,'Figure Annotation');
uisetfont(h,'Revise Font')
```

下面的表达式生成两个按钮，然后把基于其中一个设置的字体性质的值设置给另外一个。

```
a1 = uicontrol('Style', 'pushbutton', ...
      'Position', [20 10 100 20], 'String', 'MIT');
a2 = uicontrol('Style', 'pushbutton', ...
      'Position', [20 50 100 20], 'String', 'BKL');
d = uisetfont(a1)
set(a2, d)
```

14. 创建警告对话框

名称：warndlg

创建警告对话框。

语法：h = warndlg('warningstring','dlgname')

描述：warndlg 显示一个包含字符串 This is the default warning string.'的名为'Warning Dialog'的对话框。在用户按了 OK 按钮后，警告对话框消失。

warndlg('warningstring') 显示一个标题为 'Warning Dialog' 的对话框，其中包含由 warningstring 指定的字符串。

warndlg('warningstring','dlgname')显示一个标题为 dlgname 包含字符串 warningstring 的对话框。

h = warndlg(...) 返回一个对话框的句柄。

举例：

表达式

warndlg('Pressing OK will clear memory','!! Warning !!')

显示了如下的对话框：

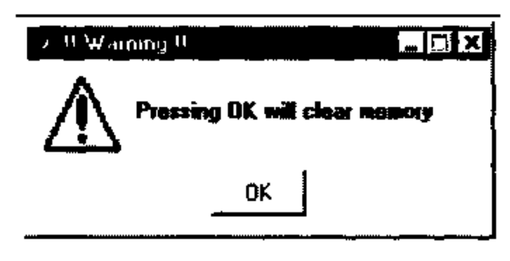


图 17-7　警告对话框

17.2　用户界面对象

1．生成菜单

名称：menu

生成菜单。

语法：k = menu('mtitle','opt1','opt2',...,'optn')

描述：k = menu('mtitle','opt1','opt2',...,'optn') 显示一个标题为'mtitle'，选项为'opt1'，'opt2'等等的菜单。menu 返回你回车的值。

举例：

k = menu('Choose a color','Red','Green','Blue') 显示下面的对话框：

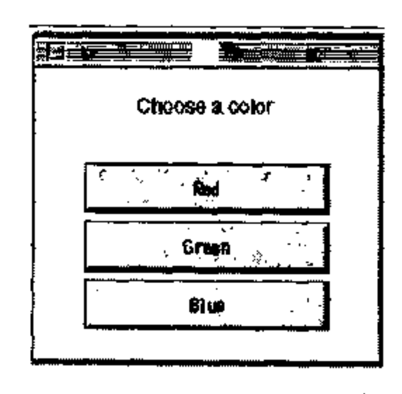


图 17-8　选择颜色对话框

输入被接受以后，可以用 k 来控制一个图形的颜色。

color = ['r','g','b']

plot(t,s,color(k))

2. 创建上下文菜单

名称：uicontextmenu

创建上下文菜单。

语法：handle = uicontextmenu('PropertyName',PropertyValue,...)

描述：uicontextmenu 生成一个上下文菜单，当用户在一个图形对象上点击右键时出现此菜单。

用户可以用函数 uimenu 产生菜单选项。菜单项按照 uimenu 声明时的顺序排列。把 uicontextmenu 的特性用于一个对象可以使上下文菜单和一个对象联系起来，并把上下文菜单的句柄指定为性质值。

下表列出了用于 uicontextmenu 对象的性质，并按照作用进行了分组。

表 17-2　　上下文菜单的属性列表

性质名称	性质描述	性质的值
控制风格和外观：		
Visible	Uicontextmenu 的可视性；	on 和 off；默认值为 off
Position	当 Visible 设置为 on 时 Uicontextmenu 的位置。	二元向量；默认值为[0 0]
对象的一般信息：		
Children	为 uicontextmenu 定义的用户界面菜单项	矩阵
Parent	对象 Uicontextmenu 的双亲	标量图的句柄
Tag	用户指定的对象表示符	字符串
Type	图形类对象	只读的字符串；默认值为 uicontrol
UserData	用户指定的数据	矩阵
控制调用程序的执行		
BusyAction	调用程序中断	Cancel 或 queue；默认值为 queue
Callback	控制行为	字符串
CreateFcn	在对象产生中执行的调用程序	字符串
DeleteFcn	在对象删除中执行的调用程序	字符串
Interruptible	调用程序中断模式	On 或 off；默认值为 on
控制对象的访问		
HandleVisibility	句柄是否受命令行和图形用户界面的影响	On, callback 和 off；默认值为 on

举例：

下述语句定义了一个和一条线有关的上下文菜单。当用户在一条线的任意位置点击鼠标时，菜单就会出现。菜单选项使用户可以改变线的类型。

```
cmenu = uicontextmenu;
hline = plot(1:10, 'UIContextMenu', cmenu);
cb1 = ['set(hline, "LineStyle", "--")'];
cb2 = ['set(hline, "LineStyle", ":")'];
cb3 = ['set(hline, "LineStyle", "-")'];
```

```
item1 = uimenu(cmenu, 'Label', 'dashed', 'Callback', cb1);
item2 = uimenu(cmenu, 'Label', 'dotted', 'Callback', cb2);
item3 = uimenu(cmenu, 'Label', 'solid', 'Callback', cb3);
```

当用户点击一条线时，下图中所示的上下文菜单就会出现：

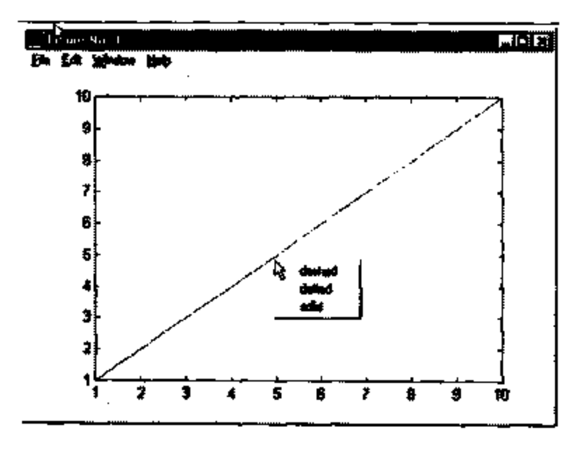


图 17-9　上下文菜单示例

对象的继承表：

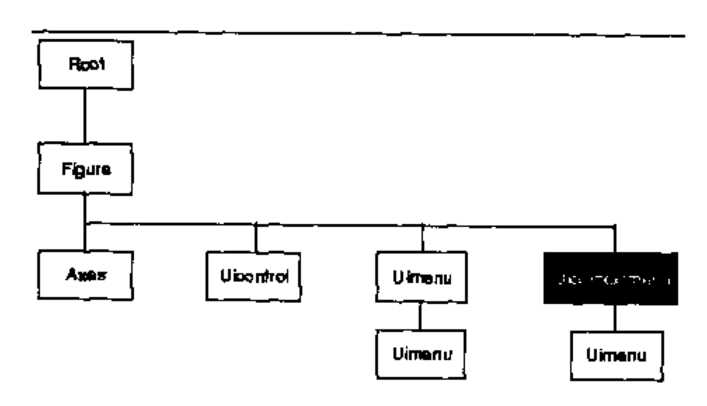


图 17-10　对象继承表

3．创建用户界面控制对象

名称：uicontrol

创建用户界面控制对象。

语法：该函数有如下种表达形式：

- handle = uicontrol(parent)
- handle = uicontrol(...,'PropertyName',PropertyValue,...)

描述：uicontrol 生成用户界面控制图形对象。你可以用各种 uicontrol 实现图形用户界面。当被选中后，大多数 uicontrol 对象执行一个预先定义的动作。MATLAB 提供了各种类型的 uicontrol，每种都有一种不同的用途：

Check boxes

Editable text

Frames

List boxes

Pop-up menus

Push buttons

Radio buttons

Sliders

Static text

Toggle buttons

Check boxes 在点中时产生一个动作。如果要为用户提供若干独立的选项时，这些设计是有用的。为了激活一个 Check box，可以在一个对象上点击鼠标。这个设计的说明会显示出来。

Editable text 框是可以使用户输入或修改正文的区域。如果用户要输入正文时可使用 editable text。

在微软的 Windows 系统中，如果一个编辑文本框有集中点时，点击菜单条不会促成 editable text 调用程序的执行。然而，在 UNIX 系统中它确实能产生调用。因此，在点中菜单条后，表达式

get(edit_handle,'String')

在 Windows 系统中不会返回编辑框的当前内容，这是因为 MATLAB 必须执行调用信息程序才能更新 String（即使屏幕上的文本串已经改变了）。这种行为和各自的平台约定相一致。

Frames 是一个图形窗口中视觉上可装入的对话框。Frames 通过视觉上相关信息的分组控制使用户界面易于理解。Frames 没有与之相应的调用程序。只有 uicontrol 可以在 frames 中出现。

Frames 是不透明的，所以在确定一个框架中的用户控制区是否由框架覆盖或是否可见时，你定义的用户界面控制命令就非常重要。堆栈的命令确定被画的对象：先定义的对象先画，后定义的对象在现有的对象上画。如果你想用一个框架来套入对象，你必须在定义对象前先定义框架。

List boxes 显示一个由特性 String 定义的选项的列表，使用户可以选择一项或者多项。性质 Min 和 Max 控制选择的模式。Value 指示被选的条目并且包含对字符串列表的索引；一个向量形式的 value 表示多个选择。在任何改变 Value 值的鼠标弹起操作后，MATLAB 计算 list box 调用程序的值。因此，为了延迟多次点击选项的动作，用户可能需要增加一个"Done"的按钮。List boxes 区别单击和双击，并把图形的 SelectionType 设置为 normal 或者在计算列表项的 Callback 时相应地设置为 open。

在点中时，pop-up menus 打开以便显示一个选项的列表（选项由 String 定义）。不打开时 pop-up menu 表示当前的选择。pop-up menus 可以用在你想给用户提供一些互斥性的高级选项时，但是不要想去消除所耗的大量的空间，这是一系列 Radio button 所必需的。你还必须为性质 String 指定一个值。

Push buttons 在点击时产生一个动作。为了激活一个 push button，可以在 push button 上面点击鼠标。

Radio buttons 类似于 check boxes，但暗含着一组相关的 radio buttons 是相互排斥的（也就是说，任何时间只有一个按钮处于点中状态）。为了激活一个 radio button，可以在对象上点击鼠标。这个图例的简要说明会显示出来。注意到你的程序可以实现 radio buttons 之间的

相互排斥性。

Sliders 在一个指定的区域内接受数字输入，使用户可以移动一个滑动条。用户可以通过点击鼠标和拖动条上的指示器来移动一个条，或者通过在一个槽和箭头上点击来实现。条的位置表示为一个数字值，它通过释放鼠标来选定。你可以为一个移动条设置最小值、最大值和当前值。

Static text boxes 显示文本的各行。Static text 典型地应用在标示其它控制时，为用户提供指示或显示和滑动条有关联的值。用户不能交互地改变静态文本并且没有办法调用它的调用程序。

Toggle buttons 是在点中和指示说明时执行调用程序的控制，值或为 on 或为 off。Toggle buttons 一般用于创建工具栏。

下标列出了所有可用于 uicontrol 对象的特性，并且按照作用进行了分组。

表 17-3　　　　用户界面控制对象的属性列表

属性名称	描述	属性值
控制风格和外观：		
BackgroundColor	对象的背景颜色	ColorSpec：默认值依赖于系统
Cdata	控制中显示的真实颜色的图像	矩阵
ForegroundColor	正文的颜色	ColorSpec：默认值为[0 0 0]
SelectionHighlight	选中时对象突出出来	On 和 off；默认值为 on
String	Uicontrol 的标签，也是 list box 和 pop-up menu 的选项	字符串
Visible	Uicontrol 的可视性	On 和 off；默认值为 on
对象的一般信息：		
Children	Uicontrol 对象没有子代	
Enable	启用或阻塞 uicontrol	On、inactive 和 off；默认值为 on
Parent	Uicontrol 对象的父代	标量图的句柄
Selected	对象是否被选	On 和 off；默认值为 off
SliderStep	滑动步的尺寸	二元向量；默认值为[0.01 0.1]
Tag	用户指定的对象标示符	字符串
TooltipString	对象功能提示的内容	字符串
Type	图形对象类	字符串（只读）；默认值为 uicontrol
Style	Uicontrol 对象的类型	Pushbutton, togglebutton, radiobutton, checkbox, edit, text, slider, frame, listbox, popupmen；缺省为 pushbutton。
UserData	用户指定的数据	矩阵
控制对象的位置		
Position	uicontrol 对象的尺寸和位置	位置矩形；默认值为[20 20 60 20]
Units	说明位置向量的单位	pixels, normalized, inches, centimeters, points, characters；默认值为 pixels
控制字体和标签		
FontAngle	字符的倾斜度	Normal, italic, oblique；默认值为 normal
FontName	字体种类	字符串；默认值依赖于系统
FontSize	字体尺寸	以 FontUnits 为单位的尺寸；默认值依赖于系统
FontUnits	字号的单位	points, normalized, inches, centimeters, pixels；默认值为 points
FontWeight	正文字符的色度	light, normal, demi, bold；默认值为 normal
HorizontalAlignment	标签字符串的对准方式	left, center, right；默认值依赖于 uicontrol 对象

续 表

属性名称	描述	属性值
String	Uicontrol 的标签，也是 list box 和 pop-up menu 的选项	字符串
控制调用程序的执行		
BusyAction	调用程序的中断	cancel, queue；默认值为 queue
ButtonDownFcn	点击调用程序的按钮	字符串
Callback	控制行为	字符串
CreateFcn	对象创建时执行的调用程序	字符串
DeleteFcn	对象删除时执行的调用程序	字符串
Interruptible	调用程序中断模式	On 和 off；默认值为 on
UIContextMenu	和 uicontrol 有关的 Uicontextmenu 对象	句柄
当前状态的信息		
ListboxTop	列表框最上部显示的字符串索引	标量；默认值为[1]
Max	赖于 uicontrol 对象的最大值	标量；默认值有赖于对象
Min	赖于 uicontrol 对象的最小值	标量；默认值有赖于对象
Value	Uicontrol 对象的当前值	标量或者向量；默认值依赖于对象
控制对象的访问		
HandleVisibility	是否可以由命令行或图形用户界面使用句柄	on, callback, off；默认值为 on
HitTest	是否由鼠标点击选择	on, off；默认值为 on

举例：

下面的表达式创建一个按键，点击它时删除当前轴。

```
h = uicontrol('Style', 'pushbutton', 'String', 'Clear',...
        'Position', [20 150 100 70], 'Callback', 'cla');
```

你可以创建一个 uicontrol 对象来改变图形的色图，只需指定一个 pop-up menu 并提供一个 M 文件名作为对象的调用信号：

```
hpop = uicontrol('Style', 'popup',...
        'String', 'hsv|hot|cool|gray',...
        'Position', [20 320 100 50],...
        'Callback', 'setmap');
```

上面对 uicontrol 的访问在菜单中定义了四个选项：hst，hot，cool 和 gray。用特性 String 指定这些选项，之间用字符"|"隔开。

在这个例子中的调用程序，setmap，是一个 M 文件的名字。它定义了一组比单纯 MATLAB 命令更复杂的指令。Setmap 包含下面的陈述：

```
val = get(hpop,'Value');
if val == 1
    colormap(hsv)
elseif val == 2
    colormap(hot)
elseif val == 3
    colormap(cool)
```

```
elseif val == 4
    colormap(gray)
end
```

特性 Value 包含一个指示被选项的数字。选项是有限的，顺序从 1 到 4。M 文件 setmap 能够得到从而检验特性 Value 的值来确定采取何种操作。

对象的继承表：

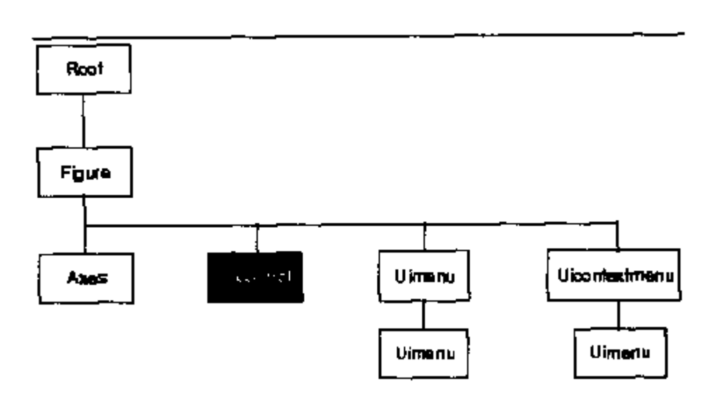


图 17-11　对象的继承表

4. 创建用户界面菜单

名称：uimenu

创建用户界面菜单。

语法：该函数有如下种表达形式：

- uimenu('PropertyName',PropertyValue,...)
- uimenu(parent,'PropertyName',PropertyValue,...)
- handle = uimenu('PropertyName',PropertyValue,...)
- handle = uimenu(parent,'PropertyName',PropertyValue,...)

描述：uimenu 产生一个菜单层和图形窗口菜单条中显示的子菜单。你也可以用 uimenu 为正文菜单创建菜单项。

handle = uimenu('PropertyName',PropertyValue,...)用指定的性质值在当前图形的菜单条中创建一个菜单并且把菜单的句柄赋给 handle。

handle = uimenu(parent,'PropertyName',PropertyValue,...)生成一个 parent 菜单的子菜单，或者在一个正文菜单 parent 中生成一个菜单项并把菜单的句柄赋给 handle。如果 parent 指代一个图形而不是另一个用户菜单对象或者用户正文菜单，MATLAB 在推荐图形的菜单条中生成一个新的菜单。

下表列出了所有可用于 uimenu 对象的特性，并且按照作用进行了分组。

表 17-4　　用户界面菜单对象的属性列表

性质名称	描述	性质值
控制风格和外观：		
Checked	菜单检查指示器	On 和 off；默认值为 off
ForegroundColor	正文的颜色	ColorSpec；默认值为[0 0 0]
Label	菜单标签	字符串
SelectionHighlight	选中时对象突出出来	On 和 off；默认值为 on

续 表

性质名称	描述	性质值
Separator	分隔线模式	On 和 off；默认值为 on
Visible	Uimenu 的可视性	On 和 off；默认值为 on
对象的一般信息：		
Accelerator	快捷键	字符
Children	子菜单的句柄	句柄向量
Enable	启用或阻塞 uimenu	On 和 off；默认值为 on
Parent	Uimenu 对象的父代	句柄
Tag	用户指定的对象标示符	字符串
Type	图形对象类	字符串（只读）；默认值为 uimenu
UserData	用户指定的数据	矩阵
控制对象的位置		
Position	相对的 uimenu 的位置	标量；默认值为[1]
控制调用程序的执行		
BusyAction	调用程序的中断	Cancel, queue；默认值为 queue
ButtonDownFcn	点击调用程序的按钮	字符串
Callback	控制行为	字符串
CreateFcn	对象创建时执行的调用程序	字符串
DeleteFcn	对象删除时执行的调用程序	字符串
Interruptible	调用程序中断模式	On 和 off；默认值为 on
UIContextMenu	和 uicontrol 有关的 Uicontextmenu 对象	句柄
控制对象的访问		
HandleVisibility	是否可以由命令行或图形用户界面使用句柄	on, callback, off；默认值为 on
HitTest	是否由鼠标点击选择	on, off；默认值为 on

举例：

这个例子生成了一个标示为 Workspace 的菜单，它的选项允许用户生成一个新的图形窗口，存储工作站变量并且退出 MATLAB。另外，它为 Quit 选项设置一个快捷键。

```
f = uimenu('Label','Workspace');
    uimenu(f,'Label','New Figure','Callback','figure');
    uimenu(f,'Label','Save','Callback','save');
    uimenu(f,'Label','Quit','Callback','exit',...
            'Separator','on','Accelerator','Q');
```

对象的继承表：

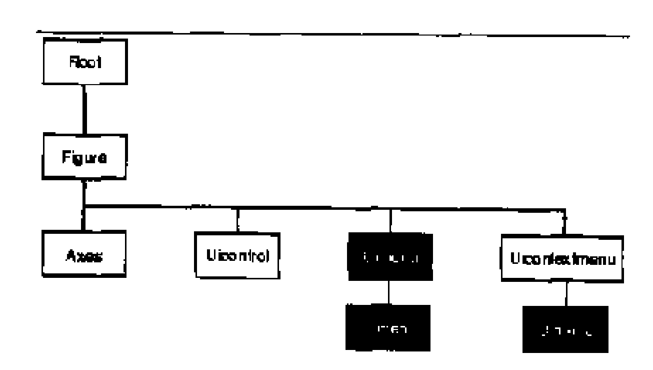


图 17-12 对象的继承表

17.3 其他函数

1. 使用鼠标拖动矩形

名称：dragrect

使用鼠标拖动矩形。

语法：该函数有如下种表达形式：

- [finalrect] = dragrect(initialrect)
- [finalrect] = dragrect(initialrect,stepsize)

描述：[finalrect] = dragrect(initialrect)在屏幕上任意位置画一个或多个矩形。n×4 的矩阵 initialrect 定义了多个矩形。initialrect 的每行必须包含下述形式的矩形的初始位置值：[左边界 底端 宽度 高度]。dragrect 把矩形的终点位置返回到 finalrect 中。

[finalrect] = dragrect(initialrect,stepsize)以 stepsize 的增量移动矩形。第一个矩形的左下角被限制在距图形左下角尺寸为 stepsize 的网格上，而其它所有的矩形都保持它们和第一个矩形的原始偏移量。[finalrect] = dragrect(...)返回释放鼠标时的最终位置。默认的 stepsize 值为 1。

举例：

下面画一个宽度为 40 个像素，高度为 80 个像素的矩形：

```
waitforbuttonpress
point1 = get(gcf,'CurrentPoint')
rect = [point1(1,1) point1(1,2) 40 80]
[r2] = dragrect(rect)
```

2. 返回正在执行回叫信息的图形对象的句柄

名称：gcbo

返回正在执行回叫信息的图形对象的句柄。

语法：该函数有如下种表达形式：

- h = gcbo
- [h, figure] = gcbo

描述：h = gcbo 返回正在执行调用信息的图形对象的句柄。

[h, figure] = gcbo 返回当前调用对象的句柄和包含此对象的图形的句柄。

MATLAB 把正在执行调用信息的对象的句柄存储在特性 CallbackObject 的根中。如果一个调用信息中断了另外一个，MATLAB 将用正在打断调用的对象的句柄来代替 CallbackObject 的值。当那个调用信息完成时，MATLAB 恢复被中断调用的对象的句柄。

性质 CallbackObject 的根是只读的，所以它的值在调用执行过程中的任何时候总是正确的。性质 CurrentFigure 的根以及性质 CurrentAxes 和 CurrentObject 的图形是用户指定的，所以在执行中可以改变，特别是在一个调用信息被另一个中断时。因此，这些函数作为正在执行调用信息的对象的指示器是不可靠的。

如果用户为任意对象的 CreateFcn 和 DeleteFcn 和图形 ResizeFcn 写调用程序时，则必须

使用 gcbo，因为那些调用信息不能更新性质 CurrentFigure 的根以及性质 CurrentAxes 和 CurrentObject 的图形；它们只能更新性质 CallbackObject 的根。

当没有调用信息在执行时，gcbo 返回[]（一个空数组）。

3. 创建橡皮框

名称：rbbox

创建橡皮框。

语法：该函数有如下种表达形式：

- rbbox
- rbbox(initialRect)
- rbbox(initialRect,fixedPoint)
- rbbox(initialRect,fixedPoint,stepSize)
- finalRect = rbbox(...)

描述：rbbox 在当前图形中初始化并画出一个橡皮框。它将块的初始矩形尺寸设置为 0，把块固定在图形的 CurrentPoint（当前点），并且从这一点开始画起。

rbbox(initialRect)以[x y 宽度 高度]的形式指定橡皮框的初始位置和尺寸，这里 x 和 y 定义了左下脚，而宽度和高度则定义了尺寸。initialRect 的单位由当前图形的 Units 性质决定，并从图形窗口的左下脚开始量度。最靠近指示器位置的块的角跟踪着指示器直到 rbbox 收到一个向上的鼠标时为止。

rbbox(initialRect,fixedPoint)指定保持固定的块的角。所用的参量都用当前图形的单位来衡量，并由图形窗口的左下脚开始量度。fixedPoint 是一个二元向量[x y]。跟踪的点是直接和由 fixedPoint 定义的落脚角相对应的角。

rbbox(initialRect,fixedPoint,stepSize)指定橡皮框更新的频率。当跟踪点超出 stepSize 个图形单位时，rbbox 重画橡皮框。默认的 stepSize 是 1。

finalRect = rbbox(...)返回一个四元向量[x y 宽度 高度]，这里 x 和 y 是块左下脚的 x 和 y 的分量，宽度和高度是块的尺寸。

rbbox 可以用于定义和修改一个矩形区域。

对于一个块的定义，initialRect 是[x y 0 0]，这里(x,y)是这个图形的 CurrentPoint（当前点）。

对于块的修改，initialRect 定义了用户要修改的矩形区域（例如一个图例）。FixedPoint 是直接和跟踪点相对应的点。

如果现在不点鼠标的话，rbbox 会马上返回。因此，你要和 waitforbuttonpress 一起使用 rbbox 以便向下点击鼠标时调用 rbbox。在释放鼠标后 rbbox 返回。

举例：

假定当前视窗为 view(2)，用当前轴的性质 CurrentPoint 来确定在数据空间中矩形的区域：

```
k = waitforbuttonpress;
point1 = get(gca,'CurrentPoint');    % 发现向下点击的鼠标
finalRect = rbbox;                   %返回图形的单位
point2 = get(gca,'CurrentPoint');    % 发现释放的鼠标
```

```
point1 = point1(1,1:2);                    % 取出 x 和 y
point2 = point2(1,1:2);
p1 = min(point1,point2);                    %计算位置
offset = abs(point1-point2);               % 计算尺寸
x = [p1(1) p1(1)+offset(1) p1(1)+offset(1) p1(1) p1(1)];
y = [p1(2) p1(2) p1(2)+offset(2) p1(2)+offset(2) p1(2)];
hold on
axis manual
plot(x,y)                     % 以数据空间的单位重画
```

4. 选择、删除、调整或复制轴图形对象和用户界面控制图形对象

名称：selectmoveresize

选择、删除、调整或复制轴图形对象和用户界面控制图形对象。

语法：该函数有如下种表达形式：

- A = selectmoveresize;
- set(h,'ButtonDownFcn','selectmoveresize')

描述：selectmoveresize 可以用作坐标系和用户控制按键函数的调用程序。执行此函数可以选择对象并允许对其进行移动、修该和拷贝。

例如，下面的表达式把当前轴的特性 ButtonDownFcn 设置给 selectmoveresize：

set(gca,'ButtonDownFcn','selectmoveresize')

A = selectmoveresize 返回一个结构数组。数组包括：

A.Type：一个包含行为类型的字符串，可以是 Select，Move，Resize 或者 Copy。

A.Handles：一个可选的句柄列表，或者对 Copy 来说是一个 m×2 的矩阵，矩阵的第一列为原始的句柄，新的句柄在第二列中。

5. 对给定的用户界面控制对象返回覆盖的字符串矩阵

名称：textwrap

对给定的用户界面控制对象返回覆盖的字符串矩阵。

语法：该函数有如下种表达形式：

- outstring = textwrap(h,instring)
- [outstring,position] = textwrap(h,instring)

描述：outstring = textwrap(h,instring)返回一个翻卷过的字符串数组 outstring，它适合于一个句柄为 h 的 uicontrol 对象。instring 是一个数组，其中每个元素包含一行正文。outstring 是在单元数组格式下翻卷过的字符串数组。输入的字符串的每个单元都被认为是一个段落。

[outstring,position]=textwrap(h,instring)返回了 uicontrol 对象以此对象本身单位衡量的推荐位置。position 考虑了多行正文在 x 和 y 方向上的区域。

举例：

下面的语句把一个内容翻卷过的字符串放入一个用户控制界面中：

```
pos = [10 10 100 10];
h = uicontrol('Style','Text','Position',pos);
string = {'This is a string for the uicontrol.',
```

```
        'It should be correctly wrapped inside.'};
[outstring,newpos] = textwrap(h,string);
pos(4) = newpos(4);
set(h,'String',outstring,'Position',[pos(1),pos(2),pos(3)+10,pos(4)])
```

6. 控制程序的执行

名称：uiwait(uiresume)

控制程序的执行。

语法：该函数有如下种表达形式：

- uiwait(h)
- uiwait
- uiresume(h)

描述：函数 uiwait 和 uiresume 分别阻滞和重新开始 MATLAB 的程序执行。

uiwait 阻滞执行直到调用 uiresume 或者当前的图形被删除。这个语法和 uiwait(gcf)一样。

uiwait(h)阻滞执行直到 uiresume 被调用或者图形 h 被删除。

uiresume(h)重新开始被 uiwait 延缓的 M 文件的执行。

在创建一个对话框时，用户应该建立一个带有调用信息的用户控制界面，使用户可以调用 uiresume 或撤消对话框。这些是函数 uiwait 中止执行后重新开始程序执行的仅有的方法。

uiwait 是使用 waitfor 命令的一种方便的方法，典型情况下和对话框一起使用。它提供方法停止生成了对话框的 M 文件的执行，直到用户回应此对话框。当它和一个模式对话框一起用时，uiwait/uiresume 中止 M 文件的执行并限制用户只和对话框交换信息。

7. 显示等待条

名称：waitbar

显示等待条。

语法：h = waitbar(x,'title')

描述：waitbar 在进行计算时可以显示计算完成的百分数。

h = waitbar(x,'title')生成并显示一个小数长度为 x 的等待条。这个等待条图形的句柄返回到 h 中。x 应该是介于 0 和 1 之间的数。每次并发地调用 waitbar 时，waitbar(x)都把条的长度延长到新的位置 x。

举例：

waitbar 的一个典型应用是在执行一个冗长计算的循环中。例如：

```
h = waitbar(0,'Please wait...');
for i=1:100, % computation here %
waitbar(i/100)
end
close(h)
```



图 17-13　等待条

8. 等待键和按钮的按动

名称：waitforbuttonpress

等待键和按钮的按动。

语法：k = waitforbuttonpress

描述：k = waitforbuttonpress 在一个图形窗口被激活时阻碍调用者的执行流程，直到函数发现用户点击鼠标或敲入一个键为止。函数在用户点击鼠标时返回 0；而在敲入一个按键时返回 1。

关于导致执行重新进行事件的其它信息可以由图形的特性 CurrentCharacter，SelectionType 和 CurrentPoint 得到。

如果图形定义了 WindowButtonDownFcn，那么在 waitforbuttonpress 之前 WindowButtonDownFcn 的调用程序的执行将返回一个值。

举例：

如果用户在一个图形窗口内点鼠标或敲入一个键时，下面的表达式将在命令窗口显示原文：

```
k = waitforbuttonpress;
if k == 0
    disp('Button press')
else
    disp('Key press')
end
```